Chapter 3

Minimalist assumptions on negation in Biblical Hebrew syntax

Chapter 2 expounded the inadequate syntactic discussions of the negative לֹא *lō*ʾ in most BH dictionaries and BH grammars. The syntactic richness of the negative לֹא *lō*ʾ preceding the verb or any other category does not figure adequately in dictionaries and grammars. Minimalist Syntax, the most recent development within Chomskyan generative syntax, provides a linguistic tool for solving the issue of the negative לֹא *lō*ʾ regarding its distribution and scope. Instead of a description of systematised products of language use, Minimalist Syntax views a grammar as a characterisation of the knowledge of a language borne in one's mind. The primary objective is to account for the production of well-formed sentences; therefore, in this sense, the focus is on syntax.

In this chapter, section 3.1.1 will provide a brief background on the major assumptions of Minimalist Syntax. Section 3.1.2 discusses the structure-building mechanism in Minimalist Syntax. Reference is also made to the different types of features with a discussion on interpretable and uninterpretable features. In section 3.1.3 the procedure of checking these interpretable and uninterpretable features is discussed, with a brief reference to the economy principles applicable to feature-checking. In 3.2 a typical BH sentence is derived exemplifying the procedure of feature-checking. Negation within generative syntax is discussed in section 3.3. References will be made to negation in a number of languages in 3.3.1. Section 3.3.2 will determine whether the negative לֹא *lō*ʾ represents a lexical or functional category. In 3.3.3 and 3.3.4 a discussion on the position of the negative לֹא *lō*ʾ in the negative phrase (NegP) and the position of the NegP

in the syntactic structure will follow. In section 3.4 a discussion on the phenomenon of scope will follow.

3.1 Minimalist syntax

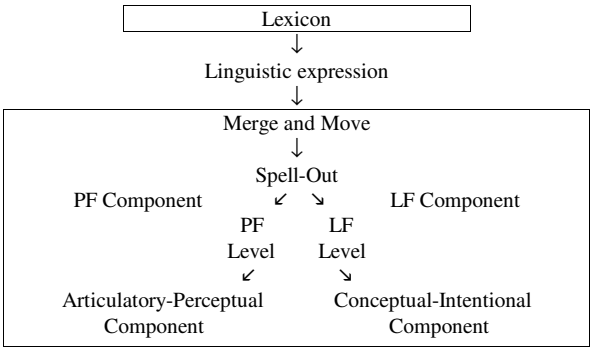
3.1.1 Assumptions of minimalist syntax

Minimalist Syntax has shared several underlying factual assumptions with its predecessors from the early 1950s, though these have proceeded to take somewhat different forms of inquiry. One of the basic assumptions of Minimalist Syntax is that the human mind/brain has a language ability, *the language faculty*, which is an autonomous system exclusively dedicated to language (Chomsky 1995: 2). The leading questions that guide Minimalist Syntax came into focus as the principles-and-parameters (P&P) theory and took shape approximately fifteen years ago. The P&P theory holds that there are universal principles and a finite array of options as to how they apply (parameters), but no language-particular rules and no grammatical constructions of the traditional sort within or across languages (Chomsky 1995: 6). Within Minimalist Syntax a *specific language* is defined as an expression of the language ability with the values of the parameters set in the functional part of the lexicon. The theory of a specific language is its *grammar* defined as the (relatively) steady state of the language ability in the mind/brain of the ideal speaker of the specific language. The initial state of the language ability is known as *universal grammar* (UG).

Chomsky (1995: 168) maintains that the language faculty is embedded in performance systems that enable its expressions to be used for articulating, interpreting, referring, inquiring, reflecting, and other actions. An expression of language can be considered as a complex set of instructions for these performance systems, providing information relevant to their functions. The performance systems with which the language faculty interacts fall into two general types: *articulatory-perceptual* and *conceptual-intentional*. Within Minimalist Syntax these systems are considered the only two mental systems that interact with the language faculty. An expression of language contains instructions for each of these systems. The interface between the computational system (the system that generates linguistic expressions) and the articulatory-perceptual system is the level of the phonetic form (PF) or soundform, while the interface between the

computational system and the conceptual-intentional system is the level of the logical form (LF) or meaning. Another standard assumption within Minimalist Syntax is that a language consists of two components: a lexicon and a computational system. The lexicon specifies the items that enter into the computational system, with their idiosyncratic properties. The computational system uses these elements to generate derivations and linguistic expressions. The derivation of a particular linguistic expression, then, involves a choice of items from the lexicon and a computation that constructs the pair of interface interpretations (Chomsky 1995: 169). The derivation of a linguistic expression begins with the selection of lexical items from the lexicon and then moves through certain steps of the computational system. At some point in the (uniform) computation to LF, there is an operation *Spell-Out* that applies to the structure already formed. *Spell-Out* strips away from a structure those elements relevant only to PF, leaving the residue of the structure to be mapped to LF (Chomsky 1995: 229). It is this prior-to-*Spell-Out* phrase marker that is pronounced (uttered). So whatever operations happen in the LF branch after *Spell-Out* are not part of the overt syntax, but are covert (Uriagereka 1998: 245). This implies that the operations after *Spell-Out* do not have a reflexion in terms of what people hear or pronounce. Against this background the organisation of the grammar can be schematised as in (1):

(1)



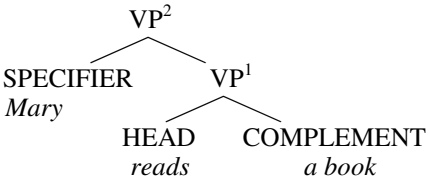
In this schema the derivation proceeds from a linguistic expression built with the elements in the lexicon to a point where (i) the linguistic expression spells out a PF representation and (ii) an LF representation is built with the linguistic expression as it exists at the point of *Spell-Out*. The

model includes – in addition to the PF and LF *levels* – PF and LF *components*, which represent the sets of derivational steps that map the relevant levels. As a consequence of this architecture, there can be no direct connection between the PF and LF levels (Uriagereka 1998:157).

3.1.2 The structure-building mechanism

The structure-building mechanism of phrases and sentences within Minimalist Syntax is known as the Generalised Transformation (GT). Minimalist Syntax claims that syntactic structures are built through generalised transformations that may insert already formed trees into trees (Marantz 1995: 359). Throughout this research, the structure of phrases and sentences will be represented in terms of a labelled tree diagram. All grammatical operations (including structure-building operations) in natural language are category-based. This means that all words in a language belong to a restricted set of grammatical categories. All phrases are formed by a process of merger whereby two categories are merged together to form a new (phrasal) category (Radford 1997: 88). Tree diagrams show how the overall sentence is built up out of constituents (*i e* syntactic units, or structural building blocks) of various types (Radford 1997: 97). Representations are built in a (strictly cyclic) bottom-up fashion. A Generalised Transformation combines two phrase markers by expanding one (the target phrase marker). This expansion takes place by adding to the target phrase marker a projection of the target phrase. This projection is binary branching and has two daughters: the target phrase marker and an empty position (Epstein *et al* 1996: 10). The building of the structure of the sentence *Mary reads a book*, *i e* a specifier, head and complement, is illustrated in (2).

(2)



Given that the computational system selects lexical items from the lexicon, it is necessary to characterise these lexical items. Items of the lexicon are of two general types: with or without content (Chomsky 1995: 54). Lexical items are considered as bundles of lexical features and are divided into two general types: substantive (or contentful) and functional (Uriagereka 1998: 61). Substantive items, on the one hand, include categories such as nouns, verbs, adjectives and prepositions. Each of these categories has their own idiosyncratic features. The heads of these categories have:

- categorial features;
- grammatical features such as ϕ-features, *i e* person, number and gender features, as well as others checked in the course of derivations;
- a phonological matrix, further articulated by the mapping to PF; and
- inherent semantic and syntactic features (Chomsky 1995: 54).

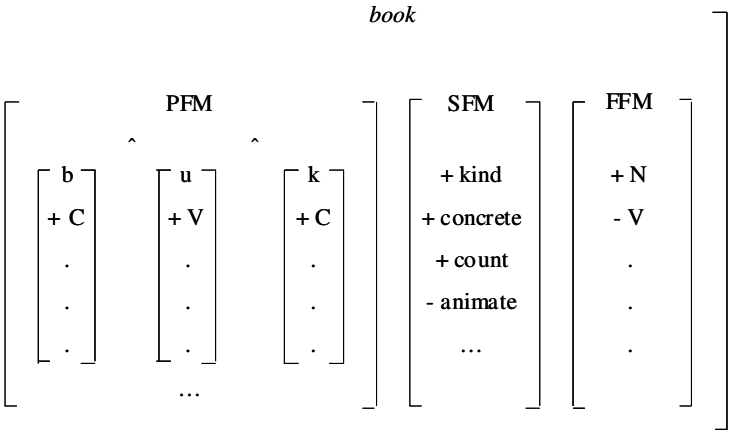
The substantive items are in the form of complete words fully inflected for case, agreement and tense. Thus, these substantive items reach the computational system with the relevant (phonologically overt) affixes and morphological features already added (Oosthuizen & Waher 1996: 38). Each lexical entry is assumed to consist of at least three sets of features: a semantic feature set, a phonological feature set, and a syntactic feature set (Epstein *et al* 1996: 8). The functional items, however, are not in the form of phonologically overt words or affixes. Functional items are abstract morphological features (case, agreement, tense, and so forth) used in the course of the derivation to check the corresponding features of the substantive items for interpretation at PF and LF. Functional items, on the other hand, include categories such as tense, complementizer, case, and so on, which is associated with inflectional morphology (Oosthuizen & Waher 1996: 38-9).

- Different types of features

In order to depict the checking of features, it has become important to differentiate among the different types of features. Uriagereka (1998: 250) notes that features are just properties of matrices: the phonological feature matrix (PFM), the semantic feature matrix (SFM), and the formal feature matrix (FFM). These three are bundled into substantive items. PF is concerned with the phonological feature matrix (PFM) encoding sensory motor instructions; LF is concerned with the semantic feature matrix (SFM); and the formal feature matrix (FFM) is relevant only to the syntax proper.

Consider the schema in (3) for the substantive item *book* taken from Uriagereka (1998: 250):

(3) Schematic lexical entry for *book*



In the above substantive item *book* the phonological feature matrix (PFM) will only apply to the PF component that will process the phonetic features. Spell-Out will split off the PFM as these features pertain to, amongst other things, the pronunciation of the word *book*. The semantic feature matrix (SFM) plays no role in the grammatical process as it pertains to purely semantic features.

- Features relevant to syntax

The formal feature matrix (FFM) refers to what Radford (1997:172) calls "grammatical features", since they determine the morphological form of items. These features can be defined as features that play a role in grammatical (morphological or syntactic) processes. The grammatical features will include categorial features, like [noun], [verb], and so forth. Grammatical features will also include ϕ-features referring to number (singular/plural), gender (masculine/feminine/inanimate) and person (which play a role in the syntax of subject-verb agreement). Grammatical features will also include case-features, *i e* features pertaining to the form of items, like nominative or accusative case, and so forth. Radford (1997: 174) further postulates that substantive items carry three different sets of grammatical features:

head-features (which determine their intrinsic grammatical properties), specifier-features (which determine the kinds of specifier that they allow) and complement-features (which determine the kinds of complement that they can take).

- Interpretable and uninterpretable features

Certain features of the formal feature matrix (FFM) enter into interpretation at LF while others are uninterpretable and must be eliminated for the sake of convergence. Thus, a crucial distinction is drawn between interpretable and uninterpretable features. Among the interpretable features are categorial features and the ϕ-features of nominals (Chomsky 1995: 277). Radford (1997: 172) makes the distinction that interpretable features at LF are features that have semantic content (and so contribute to the determination of meaning), whereas others are uninterpretable at LF (in that they have no semantic content and make no contribution to meaning). Radford (1997: 172) distinguishes between interpretable and uninterpretable features on the basis of the example in (4).

(4) She has gone

The fact that *she* is a third person singular expression plays a role at LF, as it refers precisely to somebody specific. By contrast, the fact that *she* is nominative does not play any role at LF. Radford explains the uninterpretability of case-features on the basis of the following:

(5) (a) They expect [*she* will win]

(b) They expect [*her* to win]

The fact that the italicised subject of the bracketed complement clause plays the same semantic role in both sentences (as the subject of the *win* clause), even though it has the nominative form *she* in (a) and the objective form *her* in (b), suggests that *case* is an uninterpretable feature. The fact that *has* in (4) is a present tense auxiliary has a role to play at LF (a sentence with the past tense auxiliary would have meant something else), but the fact that *has* is third person singular seems to play no role at LF (it is simply a consequence of the grammatical requirement for *has* to agree with its subject *he*). Likewise, the fact that *gone* is a participle seems to have no role to play at LF (but rather is a consequence of the

fact that *have* requires a complement headed by a verb in the +n participle form). From this discussion it is evident that certain features are interpretable at LF, while others are uninterpretable. There is a sort of criterion for validating legitimate PF and LF representations. This is called the principle of Full Interpretation. A linguistic representation satisfies Full Interpretation at LF if it consists entirely of legitimate LF objects. At the same time, a linguistic representation satisfies Full Interpretation at PF if it consists entirely of legitimate PF objects (Uriagereka 1998: 98). In terms of the principle of Full Interpretation, LF should only contain interpretable features; it then follows that uninterpretable features must somehow be deleted in the course of the derivation by means of checking. What counts as a possible derivation and a possible derived linguistic expression? A derivation *converges* if it yields a legitimate linguistic expression and *crashes* if it does not; a derivation *converges* at PF if it displays a legitimate phonetic form and *crashes* at PF if not; a derivation *converges* at LF if it displays a legitimate semantic form and *crashes* at LF if not (Chomsky 1995: 171). Legitimate entities are entities that form an interpretable representation on a specific level. Given this distinction on features and the necessity of checking uninterpretable features in order for the derivation to converge at LF, checking has then to be applied.

3.1.3 The procedure of feature-checking

Checking implies the licensing of the categories, *i e* the features of the substantive categories must be checked against the features of the selected functional categories. The procedure of checking implies that the specifier-features of a head are checked against the head-features of its specifier, referred to as specifier-head agreement;[28] likewise, the complement-features of a head are checked against the head-features of its complement. If there is compatibility between checker and checked in respect of a given feature, the relevant specifier or complement-feature is erased, because specifier and complement-features are uninterpretable, and the corresponding head-feature is erased if purely formal and so uninterpretable (but is not erased if interpretable) (Radford 1997: 175).

28 Cf Haegeman 1994: 130.

- Selection of functional items

The selection of the functional items necessary for feature-checking will depend on the nature of the substantive items selected from the lexicon. If a verb, for example, is inflected for past tense, the computational system must select a corresponding functional feature against which the tense-feature of the verb can be checked. Successful checking means that the relevant feature of the functional category is deleted. The morphological features of functional items are central in the operations of the computational system, but do not play any role at the PF and LF interface levels. These features must be deleted prior to PF and LF. If this deletion does not occur the feature will be visible on the interface levels, leading to the crashing of the derivation (Oosthuizen & Waher 1996: 39). These features, then, if potentially visible at the interfaces, must be deleted prior to PF and LF. If there is a failure to eliminate morphological features prior to an interface a derivation will crash (fail to converge) at that interface (Marantz 1995: 363).

- *Move* as feature-checking operation

Another feature-checking operation is now introduced, *viz Move*. Within Minimalist Syntax the operation *Move* is morphology-driven, implying then that there must be feature checkers in the targeted category. The sole function of these feature checkers is to force movement, sometimes overtly (Chomsky 1995: 278). There is only one generalised relation that allows one element to license another, by checking off the latter's features. In general, something moves to a *checking domain*. No element could have started in the checking domain. Rather, it comes from lower within the structure – from a domain where its basic semantic relations are encoded (Uriagereka 1998: 302). While the Generalised Transformation deals with the introduction of newly selected items into the derivation, *Move* deals with items already in the phrase marker and moves them to another position in the phrase marker. In most of its applications, *Move* is essentially a substitution operation. It selects an item, targets a category in the phrase marker and substitutes the selected item into the specifier position of the targeted category leaving behind a trace (Ouhalla 1999: 406). Certain features will be checked prior to Spell-Out, in the overt syntax, while others will be checked after Spell-Out, in the covert syntax.

- Economy principles applicable to feature-checking

It is within the spirit of Minimalist Program that derivations should be as economical as possible. Thus, no superfluous rule applications are involved in the derivation of linguistic expressions. Within Minimalist Syntax a principle of *least effort* is maintained (Chomsky 1995: 150). The analogous principle for representations would stipulate that, just as there can be no superfluous steps in derivations, so there can be no superfluous symbols in representations. This is the intuitive content of the notion of Full Interpretation, which holds that an element can appear in a representation only if it is properly licensed (Chomsky 1995: 151). On the LF side, the principle of Full Interpretation might rule out the presence of excess constituents in a structure. On the PF side, Full Interpretation might reject representations containing symbols with no phonetic realisation (Marantz 1995: 353). Zwart (1996: 306) calls this the *economy of representation*, summarised in (6):

(6) Use as few symbols as possible in the output of a derivation

The three major economy principles assumed in Minimalist Syntax are Shortest Move, Procrastinate and Greed.

- Shortest move

Shortest Move is the most technically specific economy principle. The basic idea is that a constituent must move to the first position of the right kind up from its source position. The application of Shortest Move needs to be relativized to the type of constituent moving and to the relevant landing site. Shortest Move should prohibit, for example, heads from skipping over any head position between the position they start in and the targeted landing site (Marantz 1995: 355). As already stated, feature-checking implies movement of a category to a target where its features can be checked. Each instance of Move has to select as a target its *closest possible* landing site.

- Procrastinate

As was noted earlier, there is a point in the computation of a grammatical representation where the derivation splits and heads towards the two interface levels, PF and LF. The second economy principle, *Procrastinate*, is a principle that prefers derivations that hold off on movement until after Spell-Out, so that the results of such movements do not affect PF. Pro-

crastinate is evaluated over convergent derivations; in effect, then, a derivation may violate Procrastinate in order to converge (Marantz 1995: 357). However, a violation of Procrastinate that *is* required for convergence is not an economy violation; one that is *not* required for convergence is an economy violation (Chomsky 1995: 374). This implies that, for convergence to occur, the principle of Procrastination may, in some cases, be violated. This is known as the *Last Resort Principle*, implying that a step in a derivation is legitimate only if it is necessary for convergence – had the step not been taken, the derivation would not have converged (Chomsky 1995: 200). Procrastinate, then, implies the following:

(7) Move as late as possible.

- Greed

The principle of *Greed* states that a constituent may not move to satisfy the needs of some other constituent; movement is motivated for selfish reasons, to satisfy the needs of the moving constituent. For example, a constituent should not move to a position in order to check off features of another constituent in a checking relation with that position; the constituent should move only to check off its own features (Marantz 1995:358). Last Resort is always *self-serving*: benefiting other elements is not allowed. Alongside Procrastinate, then, there is the principle of Greed: self-serving Last Resort (Chomsky 1995: 201). Greed, then, implies the following:

(8) Move α only if movement contributes to licensing of α.

Zwart (1996: 307) groups Procrastinate and Greed together under the label *Inertness*:

(9) Move as little as possible.

3.2 An illustration of a syntactic derivation

Having discussed the procedure in the building of a linguistic expression, the different lexical categories that are combined to build a linguistic expression, the mechanism of the Generalised Transformation, the procedure for feature-checking and economy principles applicable to feature-checking, this entire procedure will be illustrated with the BH example in (10):

(10) Gen 21[4]

וַיָּמָל אַבְרָהָם אֶת־יִצְחָק

wayyāmol ʾabrāhām ʾet-yiṣḥāq
and-circumcised-he abraham acc-isaac
And Abraham circumcised Isaac

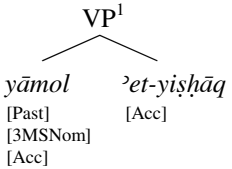
The derivation of sentence (10) begins with the selection of the substantive items, the verb *yāmol* (to simplify the derivation the conjunction וְ *wᵉ* will be left aside) and the nouns *ʾabrāhām* (subject) and *ʾet-yiṣḥāq* (object), each fully inflected with its particular morphological features (case,[29] tense and agreement). The verb *yāmol* and the nouns *ʾabrāhām* and *ʾet-yiṣḥāq* carry the following head-, specifier- and complement-features as in (11).

(11)

	ʾabrāhām	*yāmol*	*ʾet-yiṣḥāq*
Head-features:	[3MSNom]	[Past]	[Acc]
Specifier-features:		[3MSNom]	
Complement-features:		[Acc]	

The noun *ʾet-yiṣḥāq* is merged with the verb *yāmol* to form VP^1 (verb phrase) as in (12):

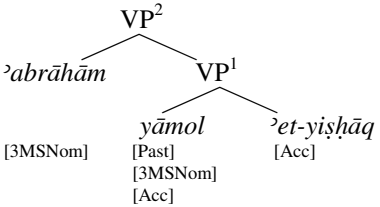
(12)



The noun *ʾabrāhām* is then merged with VP^1 to form VP^2 as in (13):

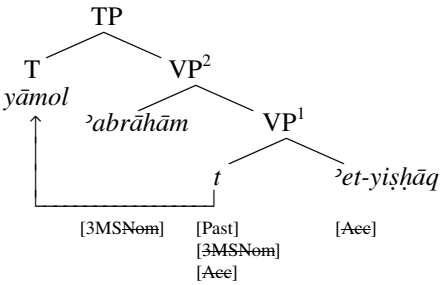
29 Cf Kroeze (2001) for an alternative view on case, especially nominative case of nominals where he argues that BH should not use the concept "nominative" as no real "cases" are found in BH. In this study *case* means abstract case.

(13)



Feature-checking in the above derivation implies that the [3MS] specifier-features of the verb *yāmol* are checked against the [3MS] head-features of *ʾabrāhām*. A perfect match is found between these two sets of features. The [3MS] head-features of *ʾabrāhām* play a role at LF. However, the [3MS] specifier-features of the verb *yāmol* play no role at LF, as they merely indicate that the verb must agree with its subject. Hence, the [3MS] specifier-features of the verb *yāmol* are deleted. The [Nom] specifier-feature of the verb *yāmol* matches the [Nom] head-feature of the subject *ʾabrāhām*, and since case-features play no role at LF, in the semantic interpretation, both [Nom] case-features are deleted. The [Acc] complement-feature of the verb *yāmol* is checked against the [Acc] head-feature of *ʾet-yiṣḥāq*. A perfect match is found and both [Acc] case-features are erased. The only feature that still needs to be checked is the [Past] tense-feature of the verb *yāmol*. To effect this checking VP^2 is merged with the head T, carrying the tense-feature, resulting in phrasal category TP (tense phrase). The verb *yāmol* is then moved to this checking position where its tense-feature is checked as in (14):

(14)



Since all the remaining features in (14) are interpretable, this derivation satisfies the principle of Full Interpretation and the derivation will converge at LF.

3.3 Negation within generative syntax

Negation in different languages has been the focus of much attention within generative syntactic research, such as that of Klima (1964), Haegeman (1995), Haegeman & Zanuttini (1991) and Ouhalla (1990). Unfortunately, no work within Generative Syntax is available on the description of negation and the NegP in BH. The only (available) source discussing the NegP in Hebrew within the parameters of Chomskyan Generative Syntax is derived from the work done by Shlonsky (1997), which focusses on Modern Hebrew and Arabic. Although Shlonsky's work might shed light on the analysis of negation and the NegP in BH, Fensham (1968:49) warns that, despite the fact that Modern Hebrew bears several similarities in vocabulary to Biblical and later Hebrew, the structure of Modern Hebrew is completely different. The following discussion will be comparative in nature, with references to negation in different languages to arrive ultimately at answers concerning the status, position and distribution of the NegP in BH. This section will provide a brief survey of the NegP in different languages, regarding the work done by Haegeman (1995), Haegeman & Zanuttini (1991) and Ouhalla (1990) as points of departure.

3.3.1 The expression of sentential negation

Some languages express sentential negation by means of one negative particle, while others exhibit multiple negative particles. As regards negation in the Romance languages, Haegeman & Zanuttini (1996:119) postulate the following three properties:

(15) (a) Sentential negation can be expressed by means of a negative marker which precedes the finite verb in linear order (eg Italian *non*, (16)(a)). Such a negative marker can be adequately described as the head of a functional category NegP and may in itself constitute the marker of sentence-negation.

(b) Alternatively, sentential negation can be expressed by means of two negative markers, one preceding and the other following the finite verb, as in French *ne* and *pas* (16)(b). In this case, the pre-verbal negative marker does not suffice for the expression of sentence-negation, and the post verbal-element is obligatory. Indeed, the pre-verbal negative marker can sometimes be omitted, in which case the post-verbal negative marker on its own may express sentential negation. If no other negative constituent is present in the clause, *pas* is normally necessary to express sentential negation.

(c) Finally, sentential negation can also be expressed by means of a post-verbal negative marker on its own, as in Piedmontese (16)(c), and in many dialects of Northern Italy, Southern France, and some Romansch dialects of Switzerland.

(16) (a) *Non* mangia. (Italian)
(b) Il (*ne*) mange *(*pas*). (French)
(c) A mangia *nen*. (Piedmontese)

'He doesn't eat.'

The question is whether BH can be described in terms of any of these three distinct means of expressing sentential negation, or whether it expresses negation in a completely unique way. To answer this question, consider a typical example of a negative sentence in BH:

(17) Gen 2[5]

כִּי **לֹא** הִמְטִיר יְהוָה אֱלֹהִים עַל־הָאָרֶץ
kî lōʾ himṭîr yahweh ʾᵉlōhîm ʿal-hāʾāreṣ
for not had-caused-to-rain-he lord god on-the-earth
…for the Lord God had not sent rain on the earth…

The sentence in (17) indicates first, that BH exhibits only one negative marker and not multiple negative markers as in French, and secondly, that in the phrasal structure the negative לֹא *lōʾ* precedes the constituent it negates, and does not follow it as in Piedmontese. As shown is (18), the following assumption may be proposed:

(18) Assumption 1
BH expresses sentential negation by means of a single negative marker, preceding the constituent it negates.

Thus, in terms of the three properties of negation in Romance languages postulated in (15), BH exhibits the following similarities and differences:

(19) (i) Like Italian, BH expresses sentential negation by means of a single negative marker.

(ii) Unlike French, BH does not exhibit two negative markers.

(iii) Unlike Piedmontese, BH does not exhibit sentential negation by means of a postverbal negative marker.

As indicated by (15) (b) and (16) (b), French exhibits two negative markers. This occurrence of multiple negative markers can lead to either *Negative Concord* or *Double Negation*. The following distinction between the Romance-type languages and the Germanic-type languages is drawn to clarify the phenomena of Negative Concord and Double Negation. The Romance-type languages exhibiting two or more negative constituents in a clause express one single instance of sentential negation, referred to as Negative Concord, as illustrated in (20)(a). In Germanic-type languages, by contrast, multiple negative constituents carry their own negative force, which means that one negative constituent will cancel the following one, resulting in Double Negation readings as in standard Dutch (20)(b):

(20)	(a)	Je	n'ai	jamais	rien	dit	à	personne.
		I	NEG have	never	nothing	said	to	nobody
		'I never told anyone anything.'						
	(b)	Ik	heb	niemand	niet	uitgenodigd.		
		I	have	no one	not	invited		
		'I did not invite no one.'						

(Haegeman & Zanuttini 1991: 233-4).

In terms of Assumption 1, BH does not exhibit multiple negation elements. Thus, with regard to Double Negation/Negative Concord, Assumption 1 suggests the following:

(21) Assumption 2
BH exhibits neither the phenomenon of Negative Concord, nor the phenomenon of Double Negation.

In summary, then, the negative לֹא *lō*ʾ expresses sentential negation with a single negative marker, which precedes the negated constituent; furthermore, BH expresses neither Negative Concord nor Double Negation.

3.3.2 The nature of the NegP

In section 3.1.2 a distinction was drawn between lexical and functional categories. In the section that follows, an examination of whether the negative לֹא *lō*ʾ represents a lexical or functional category will be made. First, however, it is necessary to provide a brief summary of the distinction between lexical and functional categories.

Lexical categories (contentives or content words) have idiosyncratic descriptive content or sense properties and include categories such as nouns, verbs, prepositions, adjectives and adverbs. Each item is a feature set. Lexical items head NP, VP, AP, and PP, and their subcategories (adverbial phrases, and so forth). The heads of these categories have (a) categorial features; (b) grammatical features such as ϕ-features and others checked in the course of the derivation; (c) a phonological matrix, further articulated by the mapping to PF; and (d) inherent semantic and syntactic features (Chomsky 1995: 54).

Functional categories (function words or functors), by contrast, serve primarily to carry information about the grammatical properties of expressions within the sentence, for instance information about number, gender, person, case, and so forth. These categories include words such as particles, auxiliaries, determiners, pronouns and complementizers (Radford 1997: 45). Functional items also have feature structure. Each functional element has certain selection properties: it will take certain kinds of complements, and may or may not take a specifier. The specifiers typically (though perhaps not always) are targets for movement. Hence, they play no independent semantic role whatsoever (Chomsky 1995: 54).

If the results on the morphology of the negative לֹא *lō'*, as discussed in Chapter 2, together with the findings of the present chapter are considered, the following conclusions can be drawn:

- The negative לֹא *lō'* does not exhibit categorial features.
- The negative לֹא *lō'* does not exhibit ϕ-features, *i e* person, number and gender.
- The negative לֹא *lō'* has no case-features, *i e* nominative, accusative or genitive, and so forth.
- The negative לֹא *lō'* has, as will be indicated in the syntactic discussion in chapters to follow, selection properties, as it will take certain kinds of complements, *i e* it is *subcategorised* for verbs, nouns, adjectives, and so forth.
- The negative לֹא *lō'*, as will be discussed in further detail below, has a fixed position in the sentence and cannot be extended. Thus, it is evident that the negative לֹא *lō'* cannot be modified by an adverb of degree.
- The negative לֹא *lō'* is a functional rather than a lexical category, because the negative לֹא *lō'* has no antonym.[30]
- The negative לֹא *lō'* is not affected by the morphological processes of inflection or derivation. In English, for example, certain

30 According to Radford (1997: 45) one test for determining whether a word is a functional or lexical category, is to see whether it has antonyms. If a word has an antonym, it is a lexical category. However, he also states that if it has no antonym, one cannot be sure whether it is a functional or lexical category. However, this test, applied with all the above-mentioned criteria, leads to the conclusion that the negative לֹא *lō'* should be considered as a functional rather than a lexical category.

> morphological processes of inflection and/or derivation may affect all lexical categories. For instance, only nouns can be inflected for the plural number (if they are *countable nouns*); only verbs can be inflected for present and past tense, the past participle, and the present participle; and only adjectives/adverbs can be inflected for the comparative and the superlative degree. Besides inflection, words may also be subject to derivational processes, which transform them from one category into another. For example the suffix *–ise*, changes a noun or an adjective into a verb, as illustrated in *generalise, naturalise, publicise,* and so forth. The above morphological processes of inflection and/or derivation cannot be applied to the negative. One cannot say, for example, *nots*, through the addition of a plural suffix. The negative cannot be conjugated for the comparative or superlative degree. In terms of derivation, one cannot add *–ise* to a negative to change its category.

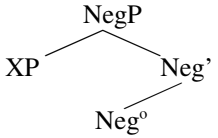
Against this background, then, it could be concluded that the negative exhibits functional rather than lexical features. On these grounds, the assumption in (22) is proposed:

(22) Assumption 3
The negative לֹא *lō*ʾ is a functor/function word, *i e* it represents a functional category.

3.3.3 The position of the negative particle in the NegP

Shlonsky (1996: 395) proposes that negation is represented on sentence level by means of a labelled XP, NegP, containing a head Neg^{o} and a specifier, as diagrammatically represented in (23):

(23)



The salient question is whether both the head and specifier positions are filled, or only the head or specifier. In terms of the diagram in (23), the question that needs to be answered for BH is whether the

negative לֹא *lōʾ* occupies the head (Neg^{o}) or XP (specifier) position in the NegP. Haegeman (1995: 127), referring to Ouhalla (1990), claims that the projection NegP is universally associated with negative sentences. As indicated by Ouhalla (1990: 189), cross-linguistic variation relates to the realization of the specifier and the head of NegP. Ouhalla suggests that superficially, languages differ in the way they express sentential negation. He refers to the following three major groups of languages. In the first group negation is expressed in terms of "a morphological category on verbs". In the second, the negation marker appears as an "auxiliary verb". In the third group the negation marker seems to bear certain adverbial properties, that is, it is "an adverb-like particle". Ouhalla leaves aside the second group and argues that the negation elements in the first and third groups are different instances of a NegP projection. Ouhalla (1990: 189) illustrates the first group with the examples in (24) (a) and (b) from Turkish and Berber, respectively:

(24) (a) John elmalar-I ser-me-di-0.
John apples-ACC like-NEG-past(TNS)-3s(AGR)
'John does not like apples.'

(b) Ur-ad-y-xdel Mohand dudsha.
NEG-will(TNS)-3ms(AGR)-arrive Mohand tomorrow
'Mohand will not arrive tomorrow.'

In both languages the negation marker appears as a constituent morpheme of the verbal complex. The third group of languages can be illustrated with the following examples from Swedish (25) (a) and (b) where the negation marker acts as an adverb:

(25) (a) Jan köpte inte boken.
Jan bought not books

(b) Om Jan inte köpte boken.
If Jan not bought books
(Ouhalla 1990: 189).

According to Ouhalla (1990: 190), the negation markers in the two language groups initially appear to differ in their categorial status; in the first group it is some sort of inflectional element, while it is an adverb

in the third. In order to identify the syntactic status of these elements, Ouhalla examines another group of languages where negation is expressed in terms of two elements which parallel the two elements discussed thus far. A representative example of this group is Standard French which expresses negation in terms of *ne*, a sort of inflectional element since it surfaces as a constituent of the verbal complex in finite clauses, and *pas*, traditionally assumed to have some adverb-like properties. French exhibits multiple negation elements in the syntactic structure, as illustrated in (26):

(26) (a) Jean ne mange pas de chocolat.

(b) Jean *ne* eats not chocolate.

In (26) sentential negation is expressed by means of the two negative elements *ne* and *pas*. Hence, one of these elements should occupy the specifier and the other the head position in the NegP. Haegeman (1995: 27) maintains that *pas* occupies a fixed position in the clause and that *ne* is a head-like element. Notice, for example, that *ne* moves with the inflected verb under subject-auxiliary inversion, as illustrated in (27):

(27) (a) Ne mange –t- il pas de chocolat?

(b) *Ne* eats he not chocolate.

Shlonsky (1996: 395) proposes that Neg^{o} dominates lexical items such as French *ne* and Italian *non* while Spec-NegP contains elements such as English *not*, French *pas*, negative operators and adverbs such as English *never*.

Ouhalla (1990: 191) summarises the variation of the NegP among languages as follows:

> Sentence-negation is generally expressed in terms of a NegP category which consists of a head element and a specifier. Variation among languages is restricted to whether either or both elements of NegP are realised lexically. In languages like Turkish and Berber the head is realised lexically while the specifier takes the form of a phonetically empty operator. In languages like German, Swedish and Colloquial French it is the specifier which is realised lexically, while the head takes the form of an abstract morpheme. Finally, in languages like Standard French both the head and the specifier are realised lexically.

According to Ouhalla (1990: 192 footnote 5) these three-dimensional cross-linguistic differences are what Zanuttini (1989) refers to as "three strategies for marking sentential negation" in the Romance languages. In languages where negation is expressed in terms of a pre-verbal element, eg Portuguese, Spanish, Catalan and Standard Italian, the head, but not the specifier of NegP, is realised lexically. By contrast, languages where negation is expressed in terms of a post-verbal element, eg the Occitan dialects and the Franco-Provençal dialects, are those where the specifier, but not the head of NegP, is realised lexically. Finally, languages where negation is expressed in terms of both a pre-verbal and a post-verbal element, *e g* Standard French and a variety of Piedmontese, both the specifier and the head of NegP are realised lexically.

The final language to be considered before examining the position of the negative לֹא *lō*ʾ in BH is Modern Hebrew. Shlonsky (1997: 12) claims that clausal negation in Modern Hebrew is implemented by the particle *lo*, which appears left-adjacent to the verb in a simple tense or left-adjacent to the auxiliary in a compound tense, as in (28) and (29), respectively:

(28) Dani lo ʔafa ʔugot.
Dani neg bake (PAST)-3MS cakes
'Dani did not bake cakes.'

(29) Dani lo haya ʔofe ʔugot.
Dani neg be(PAST)-3MS bake (BENONI)-MS cakes

The adjacency requirement of *lo* and the verb is absolute: no adjunct or parenthetical expression may separate them, as shown by the unacceptability of the sentences in (30) to (31), which should be compared with those in (32) to (33), where the same intervening elements can occur between the subject and the verb in both affirmative and negative clauses:

(30) *Dani lo kanir ʔe ʔafa ʔugot.
Dani neg apparently bake(PAST)-3MS cakes
'Dani apparently didn't bake cakes.'

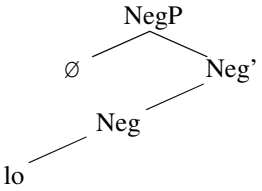
(31) *Dani lo ləda ʔati ʔafa ʔugot.
Dani neg in opinion-1S bake (PAST)-3MS cakes.
'Dani, in my opinion, didn't bake cakes.'

(32) Dani kanir ʔe (lo) ʔafa ʔugot.
Dani apparently (neg) bake (PAST)-3MS cakes.
'Dani apparently baked (didn't bake) cakes.'

(33) Dani, ləda ʔati, (lo) ʔafa ʔugot.
Dani in opinion-1S (neg) bake(PAST)-3MS cakes.
'Dani, in my opinion, baked (didn't bake) cakes.'

Shlonsky (1997: 13) claims that a straightforward way to characterise the inviolability of the cluster formed by the negative particle and the verb is to treat them as the product of syntactic incorporation of two X^o elements. This, in turn, entails that *lo* is a head – more precisely, the head of NegP. Shlonsky proposes that the Hebrew NegP consists of an overt head, *lo*, and a silent specifier, as illustrated in (34):

(34)



This analysis, however, is not without problems. According to Shlonsky (1997), the fact that no adjunct or parenthetical expression can separate *lo* and the verb indicates that both the verb and the negative *lo* occupy head positions. However, the fact that intervening elements lead to ungrammaticality, as indicated in (30) and (31), does not necessarily imply that *lo* fills the head position of the NegP. Suppose that *lo* fills the specifier position of the NegP. On such an analysis, too, no intervening phrasal constituents such as adjuncts or parenthetical expressions would be able to occur between the verb and the negative *lo*. In short, then, the

negative *lo* could fill either the specifier or the head position of NegP and in both positions intervening elements would be barred from occurring between *lo* and the verb. As Shlonsky (1997: 13) points out, however, there is another consideration that provides support for the hypothesis that *lo* fills the head of NegP in Modern Hebrew. This concerns the fact that *lo* is carried along with the verb when the latter raises to the complementiser position, as in the inversion example in (35):

(35) meʔolam lo ta ʔam Dani xacil kol kax bašel.
Never neg taste (PAST)-3MS Dani eggplant so ripe
'Never has Dani tasted such a ripe eggplant.'

If *lo* were to fill the specifier position, it need not move with the verb when the latter raises to the complementiser position. The reason for this is that a specifier (or the phrase of which it forms the specifier) cannot move to a head position. This restriction is expressed by the *Head Movement Constraint* that states that a head can only move to another head position. This implies that a specifier cannot move to a head position. Thus, *lo* must initially fill the head position of NegP, and must subsequently be incorporated into the verb, in order for the relevant head-to-head movement to occur.

Against this background on the variation among languages concerning the NegP, negation in BH will now be discussed. In terms of Assumption 1, BH does not exhibit multiple negation. The immediate question now is whether the negative לֹא *lōʾ* occupies either the specifier or head position of NegP. Consider in this regard the example in (17), repeated here as (36), as well as the example in (37):

(36) Gen 2[5]

כִּי לֹא הִמְטִיר יְהוָה אֱלֹהִים עַל־הָאָרֶץ
kî lōʾ himṭîr yahweh ʾᵉlōhîm ʿal-hāʾāreṣ
For not had-caused-to-rain-he lord god on-the-earth
…for the Lord God had not sent rain on the earth…

(37) Gen 27[2]

וַיֹּאמֶר הִנֵּה־נָא זָקַנְתִּי לֹא יָדַעְתִּי יוֹם מוֹתִי׃
wayyōʾmer hinnê-nâ zāqantî lōʾ yādaʿtî yôm môtî
And-said-he behold-now grew-old-I not know-I day death-my
Isaac said, "I have become old and don't know the day of my death…"

In both (36) and (37) the negative לֹא *lōʾ* immediately precedes the verb. Given Shlonsky's proposal for Modern Hebrew, one could argue that the verbs in both cases fill head positions and that the negative לֹא *lōʾ* also fills head positions, namely the head position of the NegP. In Chapter 4 it will be argued that examples where the negative לֹא *lōʾ* precedes verbs, as in (36) and (37), are considered as sentential negation. Hence, the verb fills a head position and the negative לֹא *lōʾ* also fills a head position of the NegP.

However, potential counter-examples to the above proposal are provided by the sentences in (38) and (39):

(38) Gen 20[5]

הֲלֹא הוּא אָמַר־לִי אֲחֹתִי הִוא [הִיא][31]
hᵃlōʾ hûʾ ʾāmar-lî ʾᵃḥōtî hîʾ
QM-not he (pronoun 3rd masc sing) said-he to-me sister-my-she
Did he not say to me, "She is my sister, …"

(39) Gen 32[29]

וַיֹּאמֶר לֹא יַעֲקֹב יֵאָמֵר עוֹד שִׁמְךָ כִּי אִם־יִשְׂרָאֵל
wayyōʾmer lōʾ yaʿᵃqōb yēʾāmēr ʿôd šimkā kî ʾim-yiśrāʾēl
and-said-he not Jacob will-be-called still name-your but Israel
Then the man said, "Your name will no longer be Jacob, but Israel,…"

Both (38) and (39) contain elements intervening between the negative לֹא *lōʾ* and the verb: a personal pronoun in (38) and a proper name in (39). In Chapters 5-7, however, it will be argued that cases where the negative לֹא *lōʾ* occurs before elements other than verbs, should be analysed as topicalisation constructions, and that topicalisation does not in fact affect

31 The *qere* reading proposes הִ א instead of הִוא, as הִוא is considered a defective form.

the head position of the negative לֹא *lō'* in the NegP. It will be argued that, even though topicalisation might occur, the negative לֹא *lō'* is still generated in the head position of the NegP. In short, then, (40) is proposed here as a working hypothesis about the head position of the negative לֹא *lō'* in BH.

(40) Assumption 4
The negative לֹא *lō'* occupies the head position of the NegP

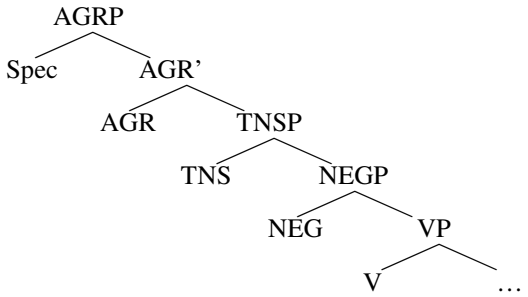
Evidence supporting Assumption (4) is that specifier positions are reserved for phrases with theta-roles, categories exhibiting person, number or gender. The negative לֹא *lō'* is not assigned any theta-role, and therefore, it cannot fill the specifier position of NegP.

3.3.4 The position of NegP in the syntactic structure

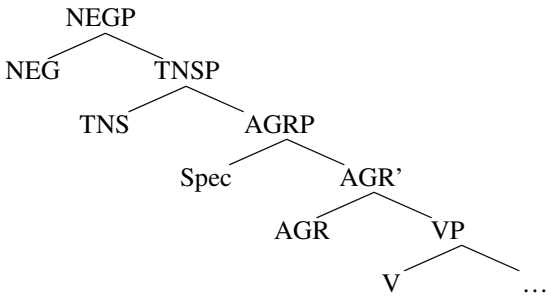
In Section 3.3.3 it was assumed that the negative לֹא *lō'* occupies the head position in the NegP. The present section focusses on the position of the NegP in the syntactic structure of BH sentences. According to Ouhalla (1990: 192), languages can be divided into two typological groups on the basis of the hierarchical position of NegP in relation to other syntactic categories. In one group NegP is closer to VP than TNS/AGR,[32] while in the other the reverse situation is found. It was illustrated in (24) (a) and (b) that Turkish and Berber are similar in that the negative is a constituent of the verbal complex. However, the two languages differ in that the negative in Turkish is inside TNS/AGR as in (41), while its Berber counterpart is outside TNS/AGR as in (42) (Ouhalla 1990: 193):

32 In the system used in this research, checking between the subject, object and verb occurs internally. The verb is then moved to TP (Ouhalla's TNSP). In the discussion below references are made to syntactic expositions using, amongst others, AGROP (agreement object phrase) and AGRSP (agreement subject phrase). These two phrases are merged for the checking of the features of the subject and the object (where applicable). In these expositions the subject moves to AGRSP, the object to AGROP and the verb to TP for the checking of the respective features. Despite the fact that this research differs from the systems explained in this section, these systems, within their respective frameworks, shed light on the position of the NegP in the syntactic derivation.

(41) Turkish



(42) Berber



The verbal complex is derived via cyclic movement of the verb to AGR in (41) and to NegP in (42). Hence, the complex derived from (41) will have NegP inside TNS/AGR, while the complex derived from (42) will have NegP outside TNS/AGR. Against this background, Ouhalla (1990: 194) proposes the following parameter for the position of NegP:

(43) The NEG Parameter

(a) NEG selects VP

(b) NEG selects TNS(P)

Turkish has value (43)(a), whereas Berber has value (43)(b). This accounts for the fact that in Turkish, NegP immediately dominates VP, while in Berber it immediately dominates TNSP.

Consider again the following Modern Hebrew sentence in (28), repeated here as (44):

(44)	Dani lo	ʁafa	ʁugot.
	Dani neg	*bake (PAST)-3MS*	*cakes*
	'Dani did not bake cakes.'		

On the basis of the surface structure of sentences like (44), Shlonsky (1997: 4-5) proposes the hierarchical order for the syntactic constituents of Modern Hebrew as in (45):

(45) AgrSP>(NegP)>TP>AspP>AgrOP>VP

This brings us to the question of the hierarchical position of NegP in BH syntactic structure. Three possibilities suggest themselves. First, it could be argued that the NegP is generated above VP, as in Turkish. Second, NegP could be generated above TP, as in Berber. A third possibility is to generate NegP above TP, as in Modern Hebrew. In connection with these possibilities, consider the BH sentence in (46):

(46) Gen 8[9]

וְלֹא־מָצְאָה הַיּוֹנָה מָנוֹחַ לְכַף־רַגְלָהּ

wᵉlō'-moṣ'â hayyônâ mānôaḥ lᵉkaf-raglāh

But-not found-she the-dove a-resting-place for-sole-of-foot-her

But the dove could find no place to set its feet…

The derivation of this example (46) will proceed along the same lines as example (10), with its derivation illustrated in (11)-(14). The different steps in this derivation will not be explained here, as the objective is to determine the position of the NegP in the syntactic structure. The features of the subject and object will be checked against the features of the verb מָצְאָה *moṣ'â* and to check the [Past] tense-feature of the verb, the latter will be moved to TP. In the surface structure the negative לֹא *lō'* precedes the verb מָצְאָה *moṣ'â*. Therefore, it is proposed that NegP is merged above TP, with the head position of the NegP filled by the negative לֹא *lō'*. Against this background the following is assumed regarding the exact structural position of the NegP in BH:

(47) Assumption 5
The NegP is generated above TP

To summarise, the following have been assumed in connection with the nature and structural position of the negative לֹא *lō:*

- The negative לֹא *lōʾ* is a functional category.
- The negative לֹא *lōʾ* fills the head position in the NegP.
- The NegP in BH is projected above TP.

3.4 The phenomenon of scope

This brings us to the question on the scope[33] of the negative לֹא *lōʾ* in BH. The discussion below does not aim to give a detailed exposition of scope in general, but merely to provide basic guidelines and assumptions for the analysis of the scope of the negative in BH. To determine the exact scope of the negative לֹא *lōʾ*, it is important to determine the syntactic relations between phrasal projections, and more specifically, the relation of the negative marker to the subsequent clause. Verkuyl *et al* (1974: 190) state that the intended semantic characteristic of the head Neg is indicated as the scope of Neg. They use the following three sentences (48) to illustrate the scope of the Neg:

33 Crystal (1985: 271) defines *scope* as follows: "Scope is a term used in syntax, semantics and pragmatics to refer to the stretch of language affected by the meaning of a particular form. For example, in English, the scope of negation normally extends from the negative word until the end of the clause; this therefore allows such semantic contrasts as *I deliberately didn't ask her* (= 'I did not ask her') and *I didn't deliberately ask her* (= 'I did ask her, but accidentally')".

(48) (a) Het is niet zo dat Jan voorzitter wordt en Peter secretaris.
It is not so that John becomes chairman and Peter secretary.

(b) Het is niet zo dat Jan voorzitter wordt en dat Peter secretaris.
It is not so that John becomes chairman and that Peter secretary.

(c) Het is niet zo dat Jan voorzitter wordt, en Peter wordt secretaris.
It is not so that John becomes chairman, and Peter becomes secretary.

Verkuyl *et al* indicate the scope of the negative in terms of the two sentences *Jan wordt voorzitter* and *Peter wordt secretaris* in (48). The latter two sentences are indicated as *p* and *q*, respectively. The differences between (48) (a/b) and (c) may be represented as follows:

(48) (a/b) NEG (*p* and *q*)

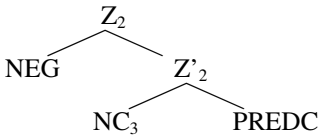
(c) (NEG *p*) and *q*

According to Verkuyl *et al* (1974: 191), the literature defines the notion of scope in two ways. The first is that of Klima (1964) and Jackendoff (1969) who define the scope of the negative by means of the notion *in construction with.* Jackendoff (1969: 218) defines the relation *in construction with* as follows:

(49) Node A is in construction with node B, if and only if the node C that immediately dominates B also dominates A.

To illustrate the definition in (49), consider the following structure in (50) adapted from Verkuyl *et al* (1974: 188):

(50)



The node NC_3 is in construction with NEG, because NC_3 is dominated by Z_2 which itself immediately dominates NEG. Jackendoff proposes that the constituents which are in construction with NEG, *i e* in the above structure Z'_2 and all nodes dominated by Z'_2, fall under the scope of the NEG.

The second manner in which scope is defined according to Verkuyl *et al* (1974: 192) is in terms of the relation *command*:

(51) Node B commands node A if and only if the lowest node Z that dominates B also dominates A.

Applied to the above diagram, (51) implies that NEG commands Z'_2 and all the nodes commanded by Z'_2.

Turning now to more recent analyses of scope, consider the example in (52) as presented by Ouhalla (1999: 155):

(52) Everyone suspects someone.

When the above sentence is pronounced with neutral intonation, it is ambiguous between at least two different meanings (or readings). On the one hand, it can have the so-called pair reading whereby each individual suspects a different individual: *Mary* suspects *John, Bill* suspects *Donald*, *Jane* suspects *Fred,* and so forth. On the other hand, it can also have the reading whereby one and the same individual is suspected by everyone: *Mary, John, Bill, Donald, Jane* all suspect *Fred.* The first reading can be paraphrased as *everyone has someone whom he/she suspects*. The second reading can be paraphrased as *there is someone whom everyone suspects*. According to Ouhalla (1999: 155), in the first reading *everyone* is said to have scope over *someone.* In the second reading the scope relation between the two quantifiers is the reverse, so that *someone* has scope over *everyone.*

As with grammatical relations in general, scope relations are expected to have a structural basis. Scope can be defined as in (53):

(53) The scope of α is the set of nodes that α c-commands in the LF representation.

The notion *c-command* is defined as in (54):

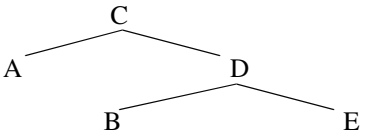
(54) α c-commands β if and only if:

(i) the first branching node dominating α also dominates β

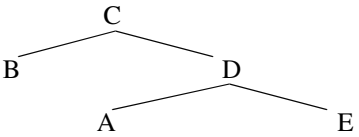
(ii) α does not dominate β.

Ouhalla (1999: 156) illustrates the notion *c-command* with the following structures:

(55)



(56)

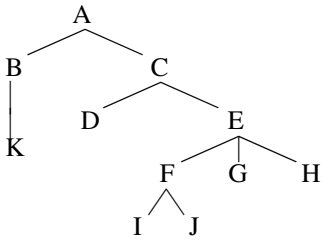


In (55) A c-commands B because the first branching node which dominates A, namely C, also dominates B. Moreover, A does not dominate B. In (56), however, A does not c-command B because the first branching node which dominates A, namely D, does not dominate B. Because A c-commands B in (55), A has scope over B, and because A does not c-command B in (56), A does not have scope over B.

The notion of *dominance* plays a key role in the definition of c-command. Let us briefly consider this notion. In a tree diagram, two relation

types exist between nodes: (a) precedence and (b) dominance. These relations can be illustrated with reference to the following structure:

(57)



Consider first the notion *precedence*: X precedes Y if and only if X and Y are linearly arranged and X appears to the left of Y. In (57), K precedes D, I, J, G, and H; J precedes G, H, but follows K, D, I. Consider next the notion *dominance*. The relation of dominance is that of containment. X dominates Y if and only if X contains Y, or if and only if Y is contained in X. ("Contains Y" means "has Y as (one of) its parts"). In (57), A dominates all the other nodes in the tree. C dominates D, E, and all the nodes that E dominates (F,I,J,G,H), but it does not dominate B, K, or A. Similarly, F dominates I and J, but does not dominate any other node. Hence, precedence is a linear relation, and dominance is a hierarchical relation. Two nodes are in either a hierarchical (dominance) relation or in a linear (precedence) relation, but not both. Thus, B and C are in a precedence relation (B precedes C), but not in a dominance relation. Furthermore, E dominates G, but neither precedes or follows the other. As regards the scope of the negative אֹל *lō*ʾ in BH, the following is assumed:

(58) Assumption 6
The scope of the negative אֹל *lō*ʾ is the set of nodes that אֹל *lō*ʾ c-commands.

3.5 Conclusion

In the first part of this chapter a brief outline was given of the major assumptions and devices of Minimalist Syntax, which forms the theoretical framework for this research. The second part focussed on the syntax of negation. The nature and structural position of the NegP in different languages

were examined in an attempt to arrive at specific assumptions about the NegP in BH. It was argued that the negative לֹא *lō'* in BH is a functional rather than a lexical category. The position of the NegP in BH syntactic structure was hypothesised to be above TP. The final part of the chapter paid attention to the phenomenon of scope, and specifically the role of c-command in determining scope relations.

The following chapters focus in more detail on the distribution and scope of negation in BH. This research moves now towards a syntactic[34] description of the negative לֹא *lō'*. Hence, the objective of Chapters 4-7 is to determine the syntactic distribution of the negative לֹא *lō'*, *i e* to determine the variety of syntactic categories adjacent to לֹא *lō'*. If one can arrive at a coherent discussion on the distribution of the negative לֹא *lō'* on sentence level, this will form the basis for discussions on the scope of the negative לֹא *lō'* in the sentence structure.

34 Crystal (1985: 300) defines *syntax* as follows in (59):

(59) "A traditional term for the study of the RULES governing the way WORDS are combined to form sentences in a language. In this use, syntax is opposed to MORPHOLOGY, the study of the word structure. An alternative definition (avoiding the concept of 'word') is the study of the interrelationships between ELEMENTS of SENTENCE STRUCTURE, and the rules governing the arrangement of sentences in SEQUENCES."