1 Introduction

This book provides an up-to-date and comprehensive linguistic analysis of the structure and meaning of the Korean language, a language that is ranked eleventh among 3,000 languages in terms of the number of speakers, being spoken by more than 77 million people, 48 million of whom live in South Korea (Ho-Min Sohn, 2001). Although typologically similar to Japanese, Mongolian, and Turkish, and thus having been in the past hypothesized to be genetically related to these languages, it is now considered a language isolate due to the implausibility of the "Altaic language family hypothesis." In addition to being a fascinating subject of study on its own, Korean has been recognized as an invaluable language to theoretical linguists by providing many counterexamples to key theoretical claims made at the forefront of modern linguistic theories. The rich and transparent inflectional morphology, morphosyntactic encodings of information structure, speech acts, register, and discourse-controlled long-distance dependency are only a few among many unique features of this language, allowing us to explore some fundamental linguistic issues that are often hard to investigate through more extensively studied English and other European languages, which often lack these features.

Although quite a few books on the Korean language are available in the publishing market today, they are either mostly descriptive or geared toward Korean learners in helping them with language proficiency. A more advanced, theoretically driven linguistics book on the Korean morphosyntax–semantics interface is currently unavailable, despite the growing community of scholars who are interested in the language. This book will discuss a broad range of empirically and theoretically important phenomena in Korean syntax and semantics that can be used by advanced undergraduates, graduate students, and professional linguists to understand the workings of the language in the framework of current thinking in the field of linguistics. The book can be used as material for both undergraduate and graduate Korean linguistics courses and graduate seminars in the departments of Linguistics, Asian Studies, Modern Languages, among others. This book may also be utilized by researchers of Linguistics as a reference book for theoretical discussions on various topics covered.

In what follows, we will introduce the syntactic and semantic tools that are used for the analysis of Korean in this book and some major linguistic characteristics of Korean.

1.1 Minimalist Syntax

1.1.1 Background

We employ the mainstream minimalist syntax for the structural analysis of Korean in this book. Chomsky fundamentally changed the field of linguistics in the mid-twentieth century, when behavioristic empiricism and positivism in science were dominant, by making statements along the following lines. Language is visible thought (mentalism). That is, language is primarily and predominantly used for thinking, which is internal. It is also used for external communication, but that function is not specific to language nor is it its dominant one. Apparently, only a part of communication is verbal. Moreover, communication is not exclusive to humans. Animals also have communication systems, but they do not appear to think like we do (Chomsky, 2005, p. 4; Jackendoff, 2002, pp. 417–418). Language is not an invented social convention, nor did it evolve over a long period of time. It was already there 50,000 years ago when humans began their journey in history. It is part of our unique biological endowment as a human species (nativism). Differences among languages are only superficial. They are mostly confined to features of lexicon and sound. Syntactic operations and principles are universal and so is the structure of human thoughts. This is the reason why all children acquire language successfully and quickly (between 18 to 30 months) despite the highly variable and incomplete environmental factors, called "poverty of stimulus."

In the 1960s, Chomsky and his colleagues launched their ambitious project of discovering the Universal Grammar (UG) and its core principles. Starting with the first two questions of balancing the **descriptive** and **explanatory adequacy**[1] of the theory, recent minimalist development focuses on the third question of why the grammar is the way it is (Chomsky, 2001b).[2] To answer this question, the strongest minimalist thesis asserts that language design is optimal, approaching a "perfect solution" to the minimal design specifications of producing legible representations or instructions at the interface levels of the sensorimotor (SM) system and the conceptual system (Chomsky, 2000,

[1] Terms in boldface are defined in the glossary at the end of each chapter.

[2] "Two immediate tasks of a theory of language are to characterize the languages attained and the shared initial stage: the task of 'descriptive adequacy' and 'explanatory adequacy,' respectively. We understand Universal Grammar (UG) to be the theory of the initial stage, and particular grammars to be theories of attained state" (Chomsky, 2000, p. 90). Descriptively adequate theories assign the correct structures to the sentences of a language, whereas explanatorily adequate theories capture the commonalities among all languages, explicating features of the possible human languages (Adger, 2003).

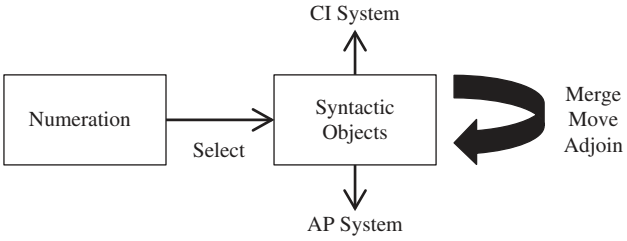

Figure 1.1. Architecture of C_{HL}

pp. 93, 96). Minimalism strives to restrict theoretical tools only to those that are absolutely necessary from the general conceptual standpoint, focusing on the efficiency and economy of the **Human Language Computational System** (in short, C_{HL}), the unique language system that generates unlimited linguistic expressions in the human mind (Chomsky, 2005).

What would a unique language system that generates an infinite number of grammatical sentences in the human mind look like? Figure 1.1 graphically represents the architecture of the C_{HL}. C_{HL} selects lexical items (called **Numeration** or **Lexical Array**) and performs syntactic operations such as **Merge**, **Move**, and **Adjoin** to build syntactic structures. The structure building is **recursive**, feeding the output back into the computational system as an input to build more complex syntactic objects. Syntactic structures consist of **Phonological Form (PF)** and **Semantic/Logical Form (LF)**, which interact with two interface systems, the **Articulatory-Perceptual system (AP)**, on the one hand, and the **Conceptual-Intentional system (CI)**, on the other.

1.1.2 Sentence and Grammaticality

Syntax and semantics concern the structure and the meaning of the **sentence**. Because the sentence is the basic unit of inquiry in formal linguistics, it is important to define it clearly, distinguishing it from other closely related concepts. The terms "propositions" and "utterances" are sometimes mistakenly used interchangeably with "sentences." A **proposition** is the message in a sentence, with all reference fixed, that can be determined as true or false. An **utterance** is an actual use of a sentence. A sentence is an abstraction of utterances that have the same form. The fact that they are independent concepts can be seen clearly in example (1), in which the proposition expressed by the sentence changes depending on who utters it and when it is uttered.

(1) Onul kipwun-i coh-ta.
today mood-NOM good-DEC
'I feel good today.'

Sentences are constructed through the shared knowledge of the speakers which they can put to use in speaking and writing. In other words, utterances are physical manifestations of sentences, the latter of which are internal, mental, and abstract entities. What we are interested in is this internal, mental, and abstract knowledge structure (**competence**) and how it is used to express propositions, rather than its physical manifestations that are variable and incomplete (**performance**).

The best way to discover the knowledge of a language is to rely on its native speakers' intuitions or grammaticality judgment. It is a common practice to flag an **ungrammatical** sentence with a star in front of it, as shown in (2). (2) is ungrammatical because it violates the basic SOV word order in Korean and contains an **honorific** agreement mismatch.

(2) *Coh-usi-ta kipwun-i onul.
good-HON-DEC mood-NOM today
[Intended] 'I feel good today.'

Native speakers may not have explicit knowledge about the grammatical rules of their language, but they never fail to understand or produce an unlimited number of well-formed sentences, expressing an infinite number of thoughts. This remarkable generative power of linguistic knowledge, which children develop fairly quickly and universally, is what minimalist syntax aims to explain.

1.1.3 Features and Interface Rules

Let us first discuss the Lexical Array or Numeration in Figure 1.1. Lexical items that assemble syntactic objects consist of features. Features are basic building blocks of syntax. Just as sentences are more abstract than utterances, features are more abstract than actual morphological forms. They are properties of words and morphemes, not actual words themselves. For example, the honorific suffix we saw in (2) has phonologically conditioned variants, *si* and *usi*. However, both forms indicate that the subject of a sentence has a higher social status than the speaker. This feature not only has an effect on the form by adding the suffix *(u)si* after the verb stem that agrees with the subject, but also on the meaning by indicating the subject's social position. Features that have effect on the meaning are called **interpretable features**, whereas those that do not affect the interpretation are called **uninterpretable features**. Interpretable features come in different types: those that reflect our basic cognitive capacities such as time, location, measure, and counting; those that concern our use of language such as speaker, hearer, topic, and definiteness; and those that are related to social and cultural aspects such as honorifics and registers (Adger, 2003). On the other hand, case feature is an example of uninterpretable features

because the nominative case does not invariably indicate a particular semantic role (e.g., the agent) of the sentence.

Once the inventory of features is decided for a language, **interface rules** between the syntactic structure and the AP, on the one hand, and the CI, on the other, need to be written. For instance, we can come up with rules in (3) for the form–meaning mapping for the honorific suffix. What a syntactic feature does is to mediate or link sound and meaning.

(3) a. Pronounce a verb specified with [honorific] by first pronouncing the verb stem and then *si* after a vowel and *usi* after a consonant.
b. Interpret a verb specified with [honorific] as indicating that the subject has a higher social status than the speaker.

It is very important to recognize that, although there is a close relationship between syntactic and semantic features, they are separate and autonomous. Think about grammatical gender not correlating with semantic gender at all, for example. One of the most important tasks of linguistics is to reconcile the mismatch between syntactic and semantic features. Minimalism, using concepts such as interpretability of features and feature deletion at the interfaces, as we will explore, explains the fundamental reasons why syntactic operations take place.

1.1.4 First Merge: Complements

Let us move on to the formation of syntactic objects by syntactic operations such as Merge, defined in (4):

(4) Merge: A constituent-building operation that joins two syntactic objects together to build a larger structure.

This syntactic operation recursively puts together smaller structures to build up larger ones. Lexical items, which are bundles of morphosyntactic and semantic features, as we have seen, serve as the smallest element in the hierarchical syntactic structure of sentences. The largest structure of syntactic and semantic inquiry is sentence, as we have mentioned. The simplest combinatorial syntactic operation, called Merge, puts two lexical items together, as shown in the tree diagram in Figure 1.2.

The objects that are merged are labeled. The label comes from the features; here, the major category feature V (for Verb) on the root node.[3] We assume that **binary branching** is universal because Merge can only combine two items at a time.

[3] Chomsky (1995) proposes **Bare Phrase Structure** that lacks labels based on the fact that the features can be read from the lexicon and thus indicating just one feature can be misleading. We will still use the major category features as labels in this book for expositional convenience.

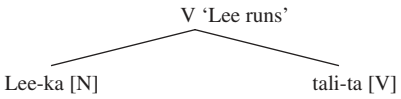

Figure 1.2. Merge

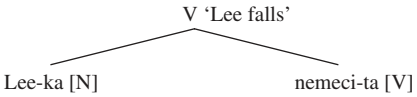

Figure 1.3. Unaccusative verb

1.1.5 Thematic Roles and C-Selection

Why does Merge take place? It is because the verb *tali-ta* 'run' cannot stand by itself but needs to combine with an **agent** argument to describe a complete event. Semantic roles that arguments play, such as agent, **theme**, etc., are called **thematic (theta, or θ) roles**. Agent arguments are those that initiate an event. The verbs that assign an agent θ-role to its argument are called **unergative** verbs, an example of which was given in Figure 1.2. In Figure 1.3, on the other hand, the verb selects a theme argument. Theme arguments are not causally involved in the event, but simply undergo a change of state. The verbs that assign a theme θ-role to its arguments are called **unaccusative** verbs.

Thematic roles concern the semantic aspects of verbs and their arguments. Because there is no one-to-one mapping between θ-roles and syntactic arguments, however, we need a separate rule for syntactic selection. Syntactic/**category selection** (or **subcategorization**) **features** determine which syntactic category a lexical item selects to merge with it. For example, *tali-ta* 'run' not only has its own category feature [V], but also the selection feature [uN] because it merges with a nominal argument. Subcategorization features are uninterpretable features, marked by u in front of N, because we cannot predict the category of a complement purely from the verb meaning. For instance, the verb *mit-ta* 'believe' can take nouns, relative clauses, and sentences as its complements, as illustrated in (5):

(5) a. Lee-ka [Kim-ul] mit-nun-ta.
Lee-NOM Kim-ACC believe-PRES-DEC
'Lee believes Kim.'
b. Lee-ka [Kim-i tolao-l kes-ul] mit-nun-ta.
Lee-NOM Kim-NOM return-REL fact-ACC believe-PRES-DEC
'Lee believes (the fact) that Kim will return.'
c. Lee-ka [Kim-i ttena-ass-ta-ko] mit-nun-ta.
Lee-NOM Kim-NOM leave-PAST-DEC-ko believe-PRES-DEC
'Lee believes that Kim left.'

The syntactic structure to which the semantic interface rules apply should consist only of interpretable features (the **Full Interpretation Principle**). Uninterpretable features must be eliminated from the syntax before the semantic interface rules apply. In fact, one of the reasons why syntactic operations apply is to eliminate uninterpretable features. An uninterpretable selection feature on a syntactic object is checked when it is a **sister** to another syntactic object which bears a matching feature. In Figure 1.4, which is a slight modification of Figure 1.2, the N feature of the subject *Lee-ka* matches the subcategorization feature of the verb [uN]. After checking, uN is deleted, which is indicated by a strikethrough on it. Unlike Figure 1.2, this structure specifies lexical and phrasal major category features. The syntactic object that selects is called the **head**. The categorical feature of the head projects to the larger phrase. In Figure 1.4, the verb is the head that projects to the whole verb phrase (VP).

1.1.6 Second Merge: Specifiers

Unlike unergative and unaccusative verbs, which take only one argument, transitive verbs select two arguments, an agent and a theme. As shown in Figure 1.5, there are two selectional N-features on the transitive verb; one is checked by merging with the object complement *Kim-ul*, but the other is checked by the merged subject NP *Lee-ka* because it is not satisfied by the first application of Merge.

In Figure 1.5, as in Figure 1.4, the root node is marked as phrasal, i.e., VP, since there are no more selectional features of V to be checked. Syntactic objects that have no category-selectional features (c-selectional features henceforth) to

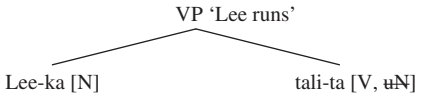

Figure 1.4. Elimination of subcategorization feature

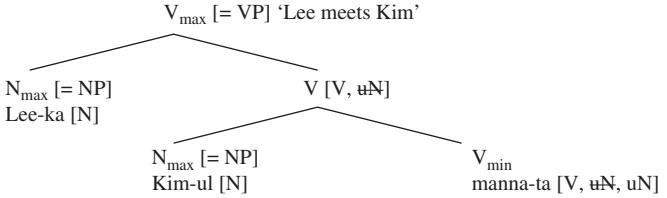

Figure 1.5. Second Merge and maximal and minimal projections

be checked are also called maximal, since they project no further. A **minimal projection** is just a lexical item. Although we have marked the complement of the head as a lexical item (e.g., N), note that in fact only **maximal projections** can be sisters of c-selecting heads. For example, as we see in Figure 1.5, the verb merges with the whole NP, not just N. If a non-maximal projection is selected, its unchecked c-selection features cannot be deleted, resulting in ungrammaticality. Finally, the different word orders arise because of different linearization properties of the head–complement structures; English is head-initial (Verb-Object) and Korean is head-final (Object-Verb). However, what is important is not the linear order but the structural relationship. Regardless of whether the verb comes at the end or the beginning in the VP, the specifier is higher than the complement in the hierarchical structure in both English and Korean.

1.1.7 Constituents and C-Command

We have observed that sentences are organized in a systematic way containing the head V, the complement NP, and the specifier NP. NPs and VPs are syntactic **constituents**, a sequence of words that behave as syntactic and semantic units. Various syntactic tests can be used to determine constituency. Take **scrambling**, rearrangements of the basic word order, as an example. In Korean, an object NP constituent can "scramble out" of the VP to the front. In this case, the whole NP must move together because it forms a constituent; an individual part of the NP, which does not form a constituent, cannot move on its own, as shown in (6).

(6) a. [Siwenha-n mwul-ul] Lee-ka masi-ess-ta.
cold-REL water-ACC Lee-NOM drink-PAST-DEC
'Cold water, Lee drank.'
b. *Mwul-ul Lee-ka siwenha-n masi-ess-ta.
water-ACC Lee-NOM cold-REL drink-PAST-DEC
*'Water, Lee drank cold.'

Native speakers' intuition about constituency is another strong piece of evidence for the systematic organization and interdependence of syntactic objects occurring in a sentence, which was modeled using the first and the second Merges.

We have been representing the syntactic structure of a sentence using tree diagrams. Such visualization allows us to see the syntactic hierarchical relations clearly. In any syntactic tree, the most important syntactic relation that holds between two nodes is that of **sisterhood**, which is built up by the fundamental operation Merge. In addition to sisterhood, there is another important syntactic relation that holds between nodes in a tree: this is the relation of

c-command (an abbreviation of **constituent-command**). We define c-command as follows:

(7) A node A c-commands a node B if and only if B is A's sister or A's sister contains B.

The new object created by Merge immediately **contains** the original object. In Figure 1.6, the highest projection VP contains the subject NP in the specifier and the intermediate V projection and its constituents, the object NP in the complement and the head V. The NP in the specifier, which is created by the second Merge, c-commands V because it is the sister of the NP and its constituents because they are contained by V. The c-command relations are indicated by the arrows. As already mentioned, structural relations such as c-command are much more important than the linear order because the sentence is not merely a string of words but a systematically organized structure based on constituency and structural dependency.

Let us observe the utility of the notion of c-command in syntactic accounts. The notion of c-command explains an asymmetric binding relation in (8). The **antecedent** *Lee-ka* must c-command the **reflexive pronoun** *caki* for a **coreferential reading**, which is marked by the same subscript/index.

(8) a. Lee$_1$-ka caki$_1$-lul piphanhay-ss-ta.
Lee-NOM self-ACC criticize-PAST-DEC
'Lee$_1$ criticized himself$_1$.'
b. *Caki$_1$-ka Lee$_1$-lul piphanhay-ss-ta.
self-NOM Lee-ACC criticize-PAST-DEC
*'Himself$_1$ criticized Lee$_1$.'

1.1.8 Adjunction

The previous sections presented the way in which the verb and its core arguments are combined using concepts such as Merge, c-selection, and θ-role assignments. In this section, we will encounter a somewhat different syntactic operation. In (9), *Lee-ka* and *Kim-ul*, two major constituents in the sentence,

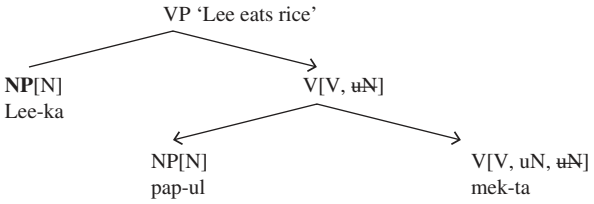

Figure 1.6. C-command

are assigned θ-roles of agent and theme by the verb *manna-ta* 'meet.' The other major constituent in the sentence *mayil* 'every day,' on the other hand, does not receive a θ-role from the verb. Semantically it has the function of indicating the time of the situation described by the rest of the sentence. That is, this constituent serves a role of **modification** rather than argument.

(9) Mayil Lee-ka Kim-ul manna-n-ta.
every day Lee-NOM Kim-ACC meet-PRES-DEC
'Lee meets Kim every day.'

The optional modification constituents are called **adjuncts**. Adjuncts are not incorporated into the sentence by Merge in the narrow sense. We assume that there is another basic operation, Adjoin. Unlike Merge, it does not need to be triggered by the head requiring an argument role to be filled. Adjoin inserts a phrasal object into another phrasal object without creating a new object. Different constituents of a sentence, then, are distinguished in terms of their structural position in the syntactic tree. **Complements** are sisters of lexical items; **specifiers** are sisters of X nodes, and adjuncts are sisters of XP nodes, as shown schematically in Figure 1.7.[4]

1.1.9 Functional Projections

In addition to lexical categories such as nouns and verbs, languages also have **functional categories** such as determiners, tense, mood, and **speech act**. Functional categories, like lexical ones, project their own phrases. For example, the past tense suffix *ess* is considered to be the head of the functional category Tense Phrase (TP), parallel to the lexical categories NPs and VPs, as illustrated in Figure 1.8. However, it would be odd to say that tense subcategorizes the VP as an argument. Instead, we will assume that a VP has an unvalued **tense feature** that needs to be valued by merging with the T head under sisterhood. This is a reasonable assumption because VPs describe events or states that are located in time.

As already observed, functional projections such as TP are not triggered by the selection features of the verb, but instead projected to host functional morphemes and contain extra semantic information such as the time of the event. To capture the different mechanism, we propose the **hierarchy of projection** in (10), following Adger (2003).

(10) Hierarchy of Projection: T > V

[4] Because different constituents are distinguished structurally, minimalists no longer employ the traditional X-bar theory, which enforces a uniform structure to all phrasal categories. In the minimalist program, if a head lacks a complement and specifier, it becomes automatically phrasal without projecting X′ and XP levels.