Korean Syntax and Semantics

EunHee Lee

Korean Syntax and Semantics

Korean has proven to be an invaluable language to theoretical linguists, providing abundant examples of, and counterexamples to, key theoretical issues at the forefront of modern linguistic theories. Exploring the Korean language from both a syntactic and semantic perspective, this book provides an up-to-date linguistic analysis of its structure, combining minimalist syntax with accompanying compositional formal semantics. EunHee Lee's detailed chapters cover the core architecture and phenomena of Korean, looking at the lexical layer, the functional layer, nominal structure, movements, and complex clauses. A broad range of empirically and theoretically important phenomena are discussed, enabling students and professional linguists alike to understand the workings of the language in current theoretical frameworks. The book also includes discussion questions, exercises, and lists of further reading to solidify the theoretical concepts, stimulate thinking, and develop the ability to analyze Korean using theoretical tools.

EUNHEE LEE is an associate professor in the Department of Linguistics at University at Buffalo, The State University of New York. Her main research areas are semantics and Korean linguistics. She is (co-)author of two books: *Korean Tense and Aspect in Narrative Discourse* (2012) and *Introduction to Korean Linguistics* (2016).

Korean Syntax and Semantics

EunHee Lee
University at Buffalo, SUNY

CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE
UNIVERSITY PRESS

University Printing House, Cambridge CB2 8BS, United Kingdom

One Liberty Plaza, 20th Floor, New York, NY 10006, USA

477 Williamstown Road, Port Melbourne, VIC 3207, Australia

314–321, 3rd Floor, Plot 3, Splendor Forum, Jasola District Centre,
New Delhi – 110025, India

79 Anson Road, #06-04/06, Singapore 079906

Cambridge University Press is part of the University of Cambridge.

It furthers the University's mission by disseminating knowledge in the pursuit of education, learning, and research at the highest international levels of excellence.

www.cambridge.org
Information on this title: www.cambridge.org/9781108417198
DOI: 10.1017/9781108265041

First published 2019

Printed and bound in Great Britain by Clays Ltd, Elcograf S.p.A.

A catalogue record for this publication is available from the British Library.

ISBN 978-1-108-41719-8 HARDBACK

Contents

Abbreviations

ACC	accusative case
BENE	benefactive
CAU	causative
CL	classifier
COM	comitative
COMP	complementizer
COND	conditional
CONJ	conjunction
CONT	contrastive
COP	copula
DAT	dative
DEC	declarative
DEF	deferential
DIR	direction
DISJ	disjunction
DPAST	double past
EVI	evidential
EXCL	exclamation
EXH	exhortative
FOC	focus
FUT	future
GEN	genitive
HEAR	hearsay
HON	honorific
IMP	imperative
INSTR	instrument
INT	interrogative
LOC	locative
MOD	modality
MOOD	mood
NEG	negation
NOM	nominative

NOMI	nominalizer
PASS	passive
PL	plural
POL	polite
PRES	present
PROG	progressive
PROM	promissive
PROP	propositive
QUOT	quotation
REL	relative clause
RESL	resultative
SC	small clause
SOUR	source
SUSPECT	suspective
TOP	topic

Romanization

Vowels and Diphthongs	Yale Romanization	Consonants	Yale Romanization
ㅏ	a	ㄱ	k
ㅐ	ay	ㄲ	kk
ㅑ	ya	ㄴ	n
ㅒ	yay	ㄷ	t
ㅓ	e	ㄸ	tt
ㅔ	ey	ㄹ	l
ㅕ	ye	ㅁ	m
ㅖ	yey	ㅂ	p
ㅗ	o	ㅃ	pp
ㅘ	wa	ㅅ	s
ㅙ	way	ㅆ	ss
ㅚ	oy	ㅇ	ng
ㅛ	yo	ㅈ	c
ㅜ	wu	ㅉ	cc
ㅝ	we	ㅊ	ch
ㅞ	wey	ㅋ	kh
ㅟ	wi	ㅌ	th
ㅠ	yu	ㅍ	ph
ㅡ	u	ㅎ	h
ㅢ	uy		
ㅣ	i		

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).

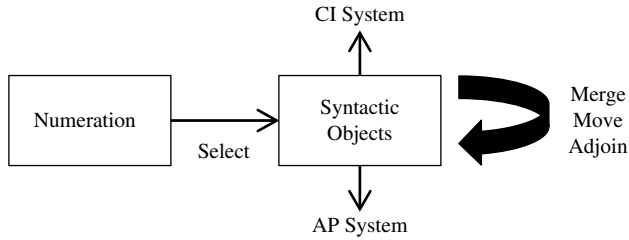

Figure 1.1. Architecture of C_{HL}

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.

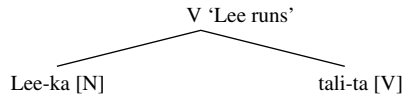

Figure 1.2. Merge

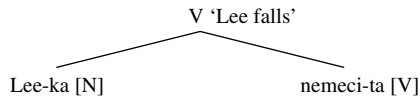

Figure 1.3. Unaccusative verb

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

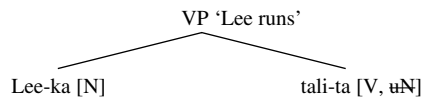

Figure 1.4. Elimination of subcategorization feature

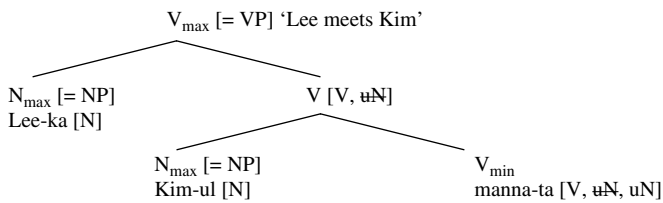

Figure 1.5. Second Merge and maximal and minimal projections

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,

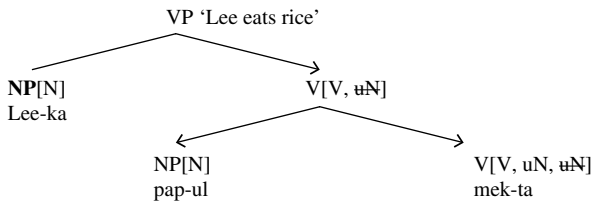

Figure 1.6. C-command

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.

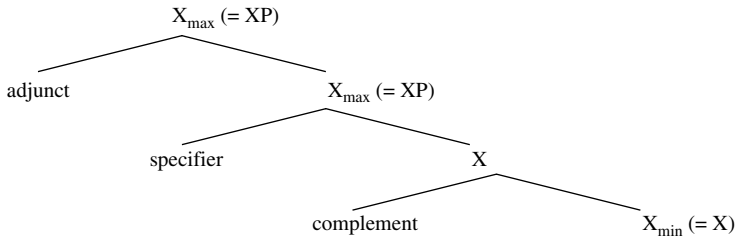

Figure 1.7. Different constituents of a sentence

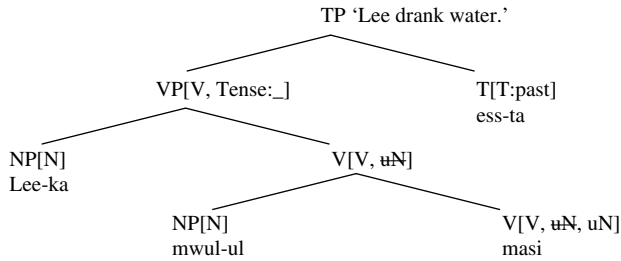

Figure 1.8. Tense Phrase

As will be discussed in detail in Chapter 3, Korean has many functional projections other than Tense Phrase, including Mood, Evidential, and Speech Act Phrases.

1.1.10 Move

Once lexical items are merged to form phrases and clauses, constituents sometimes move from their original place to create non-local dependency relations. We have already observed an example of Move, namely, scrambling, when we discussed constituency. Chomsky (2000) takes movement simply as a particular type of Merge. He refers to merge operations that select lexical items and merge them with some other constituent as **External Merge (EM)**, and to movement operations by which an item in an existing structure is moved to a new position as **Internal Merge (IM)**. Contrary to the prevalent prejudice against IM, it is in some sense simpler than EM because it requires much less search in the Lexical Array (Chomsky, 2013).

Why do constituents move? Motivations for IM are different for different types of movement. It is commonly assumed that there are two types of movement, namely **argument- (A-) movement** and **non-argument (A′) movement** (Chomsky, 1981). To understand the difference between the two types

of movement, reference needs to be made to layers of the clause structure. A clause consists of three kinds of structural layers (Rizzi, 1997). First, the lexical layer is headed by a verb where all theta assignment is made. Second, the inflectional layer (i.e., TP projection) licenses argumental syntactic features such as case and agreement. Third, the complementizer layer hosts various operators such as **wh-phrases**, and focused and topicalized elements, among others, at least in languages like English. Generally speaking, the inflection layer coincides with the A-movement and the complementizer layer with the A′-movement. An example of A-movement is the movement of a subject to Spec of TP from the VP-internal position (Koopman and Sportiche, 1991) driven by the **Extended Projection Principle (EPP)** or **edge feature**, which demands an extension of a phrasal projection with a specifier (in short, Spec) on the edge. For example, it requires a TP project to have the subject in the Spec, TP position syntactically regardless of whether the subject is assigned a θ-role by the verb. The movement occurs because the EPP feature in T is a **strong feature** that requires local feature checking. Non-local feature checking is also possible when the relevant feature is weak. Therefore, in the minimalist framework, the feature strength explains why certain movement occurs, formulated as the **Feature Strength Parameter**.

Internal Merge is also evoked by the need to rearrange the linguistic expression for scopal or discourse-related reasons. This constitutes A′-movement involving **force/speech act**, **topic**, **focus**, etc. Typical examples of A′-movement are **topicalization** and **wh-movement**. Wh-movement is obligatory in English, implying that the wh-feature in the Complementizer is a strong feature in this language. On the other hand, wh-words remain in situ in Korean, suggesting that the wh-feature in Korean is weak. We will discuss movement/Internal Merge in detail in Chapter 5.

1.1.11 Phase

As we have mentioned, syntactic objects consist of the Phonological Form (PF) component and the Semantic/Logical Form (LF) component. The PF syntactic component is sent to the Articulatory-Perceptual (AP) interface to be spelled out. Because unvalued features are illegible to the AP interface, they must be valued by syntactic derivations; otherwise the derivation crashes (Chomsky, 2005). Likewise, the LF syntactic component is sent to the CI interface. Because uninterpretable features cannot be processed in the CI interface, they must be deleted in the course of a semantic computation and become only accessible to the PF component. Computation succeeds if it converges at CI and AP, meaning if it generates legible forms at both ends. If not, it crashes.

The transfer of syntactic object to the interfaces that we have just described does not occur all at once but in stages or **phases** to make complex derivations

manageable.[5] Chomsky (2001a) suggests that there are two phases in a clause: the Complementizer Phrase (CP) and the vP (read "small vP," a vP with an agent or experiencer external argument, which will be discussed in Chapter 2). His rationale for taking CP and vP as phases is that phases are "propositional" in nature, CP representing a complete clausal complex (including a specification of force), and vP representing a complete thematic (argument structure) complex (including an external argument). To achieve an optimal computation, once a phase is created and transferred to interfaces, later operations cannot refer back to what has already been completed, giving rise to **cyclic operations**. An important consequence of strict cyclicity is that after a phase is transferred to the interfaces, only some of its contents remain accessible to subsequent syntactic operations. In a structure with a phase head and its complement, the interior of the phase is the complement part, whereas the edge is the head together with anything merged to it. No further modification can occur to the interior complement, whereas elements of the edge are still available for further modification in the next higher phase, formulated as the **Phase Impenetrability Condition** (PIC) (Chomsky, 2000).

1.1.12 Summary

This section provided a condensed overview of minimalist syntax using Korean examples. We have discussed foundational assumptions or beliefs of minimalism such as nativism and mentalism, its goal of explaining why the human language is the way it is, and universal derivational mechanisms of Merge. We also introduced some important theoretical concepts such as feature; structural relations in a syntactic tree such as sisterhood and c-command; names of constituents created through first and second Merges and Adjoin, namely, complements, specifiers, and adjuncts; and cyclic syntactic derivation in phases to explain the economy and efficiency of C_{HL}. Minimalism provides a highly abstract picture of human language, which is necessary to explain how languages are learned, or more accurately "activated," so quickly in children's minds, parallel to other physical and cognitive developments. It views language as being highly systematic and derived from a small number of basic axiomatic principles and minimal derivational operations while possessing the internal capacity of generating an infinite number of sentences in a combinatory manner with unlimited expressive power.

[5] Recently much attention has been paid to processing issues, which concern performance and have not been considered as part of the speaker's knowledge of the language, i.e., competence. Chomsky's claim that locality is forced by the need to minimize search suggests a processing explanation. The implication is that the Language Faculty can only hold a limited amount of structure in its active working memory, processing only limited amounts of structure at one time. The derivation of expressions proceeds by phase, so that syntactic structures are built up one phase at a time to help reduce the computational burden.

1.2 Formal Semantics

1.2.1 Truth Conditions

We will employ mainstream formal semantics for the semantic analysis of Korean in this book since it provides a well-developed tool for compositionally computing meaning from the syntactic structure of a sentence. Formal semantics is built on the foundational assumption that knowing the meaning of a sentence equals knowing its **truth-conditions**, the conditions under which the sentence is true or false. For example, to know the meaning of (11), one must know what the world should be like in order for it to be true. In this instance, it must be the case that an individual named *Lee* is performing the action of running at the time of speech. The string of words put together in (11) describes a situation in the world and knowing when it holds true is considered to be an important component of a native speaker's intuition concerning linguistic meaning.

(11) Lee-ka tali-n-ta.
Lee-NOM run-PRES-DEC
'Lee is running.'

By uttering sentences such as (11), the speaker conveys information to the addressee about ourselves and the world. Truth-conditional semantics focuses on this information-exchanging aspect of language, seeing semantics as a matter of the connection between language and the external world talked about. Therefore, it is also called "denotational" or "referential" semantics, because meaning connects linguistic forms to what they denote or refer to in the external world.

Formal semantics takes the **denotation** of the name *Lee* in (11) to be the actual flesh and blood person Lee. The denotation of the whole sentence, however, is less obvious. It is more complex than a single individual and also more abstract. Formal semantics takes the denotation of a sentence to be its **truth-value**, which is either true, represented by the number 1, or false, represented by the number 0.

1.2.2 The Principle of Compositionality

As we have emphasized, the linguistic system of a language can generate an infinite number of sentences via a finite number of syntactic rules. From a semantic point of view, this means that such a linguistic system enables us to express a countless number of thoughts. We humans understand and produce sentences we have never heard or said before every day. For instance, even if we do not know who Lee is and have never heard (11) before, we have no problem understanding it. One simple explanation for this remarkable feat is that we are able to "compute" the meaning of sentences from the meaning of their

parts (words) and the way they are put together (syntactic rules). This requires knowledge of the meaning of words and knowledge of semantic compositions corresponding with syntactic operations. Assuming that a language has only a finite number of words and syntactic rules, combining them in various ways as well as applying the rules recursively will generate an infinite number of sentences. This explains the productivity of language in spite of our limited memory. It would be implausible to assume that we learn language by memorizing all the sentences we hear. Moreover, if there is a tight one-to-one correspondence between syntactic rules and semantic interpretations, the latter will simply ensue from the former. This is called the **Principle of Compositionality** or **Fregean Principle**.

(12) The Principle of Compositionality: The meaning of a complex expression is determined by the meaning of its parts and the syntactic rules by which they are combined.

Formal semantics is faithful to this principle. Doing semantics, in essence, is developing a theory of meaning composition, focusing on how sentences are built up out of their parts and figuring out the contribution of each part to the meaning of the whole.

1.2.3 Sets and Functions

In truth-conditional semantics, the mathematical concepts of **sets** and **functions** play a crucial role in the semantic interpretation. Before we show how the meaning of simple sentences is compositionally derived, let us offer a brief introduction of sets and functions. A set is a collection of any objects, either finite or infinite. The objects in a set are called the **elements** or **members** of that set. Sets are defined by either listing their members or by abstraction, i.e., by stating a property that an object must have to qualify as a member. For a concrete example, let A be the set of all people. To define this set by listing, the names of its members are put inside curly brackets separated by commas, as in (13a). In (13a), *Lee*, *Kim*, and *Park* are names of people in the world (or in our universe or domain of discourse). To define it by abstraction, a condition or a property that all the members of the set share is given after a colon following the first occurrence of a **variable**, which stands for no particular objects but rather indicates what the property applies to, as in (13b). (13b) is read "the set of all x such that x is a person."

(13) a. A = {Lee, Kim, Park, ...}
b. A = {x: x is a person}

Set membership is expressed by the symbol ∈: "a ∈ A" means a is an element (or member) of the set A, and "a ∉ A" means that a is not an element

of A. In (13), Lee ∈ A, but Choi ∉ A. It is not the case that every set has to have a member. There is a set with no members. The symbol for an **empty set** is Ø.

Given two sets A and B, we can perform set-theoretic operations on them. First, if one set is completely inside the other, we call it **set inclusion**. ⊆ is the symbol for this relation: "A ⊆ B" means set A is a **subset** of set B and B is a **superset** of A, which holds when all of the members of the set A are also members of the set B. Second, two sets can intersect with each other. **Intersection** of set A and set B, "A ∩ B", includes common members of A and B. **Union** of set A and B, "A ∪ B", includes all members of A and B. **Complement** "A – B" includes all members of A which are not in B. Finally, two sets can be **disjoint** if they have no common member.

If we have two objects x and y, we can form an **ordered pair** <x, y>. Ordered pairs and sets are different. Since two sets with identical members are themselves identical, the order does not matter, i.e., {a, b} = {b, a}. This is not the case with ordered pairs, i.e., <a, b> ≠ <b, a>. A function is a set of ordered pairs in which the second member of each pair is uniquely determined by the first. Every function has a **domain** and a **range**. When A is the domain and B is the range of f, we say that f is from A and to B and write it as "f: A → B". Whenever f is a function and x is an element of its domain, there is a unique y in its range such that <x, y> ∈ f. Let us call the first member of an ordered pair an **argument** and the second member a **value**. We can think of a function as a machine that, when fed an argument as its input, gives a value as its output by operating on the argument.

Just like sets, a function can be defined by listing its elements, as in (14), in the form of a table, as in (15), or in words, as in (16). In (15), the left column lists the domain and the right column lists the range and an arrow points from each argument to the value it is mapped to. In (16), the symbol f(a) is read as f applied to a or the value of f for the argument a. In this instance, it is b, i.e., f(a) = b. F is a function because, although both a and c are mapped to the same element in the range (that is, b), the value of each member of the domain (a, c, and d) is uniquely determined. By contrast, a set of ordered pairs {<a, b>, <a, c>} is not a function, because a is mapped to both b and c, and we cannot determine the unique value of a. In such a case, it is called a **relation**, but not a function.

(14) f = {<a, b>, <c, b>, <d, e>}

(15) $$f = \begin{pmatrix} a \to b \\ c \to b \\ d \to e \end{pmatrix}$$

(16) f is a function with domain {a, c, d} such that f(a) = f(c) = b and f(d) = e

Just as we define sets by specifying the conditions on their members, we can define functions in a similar way, especially for those with infinite domains. (17) is an example. The function in (17) adds 1 to the set of natural numbers. It will map 1 to 2, 2 to 3, 3 to 4, and so on and so forth.

(17) f: N → N (N is the set of natural numbers) for every x ∈ N, f(x) = x + 1.

1.2.4 Interpretation of Intransitive Sentences

Now that the basic tools of sets and functions are laid out, we will introduce the topic of semantic composition for the interpretation of simple sentences. In formal semantics, semantic composition is **functional application**, which is expressed by the symbol f(a), meaning f applied to a or the value of f for the argument a. The meaning of a simple intransitive sentence such as (11), which is repeated in (18), is computed as follows:

(18) Lee-ka tali-n-ta.
Lee-NOM run-PRES-DEC
'Lee runs.'

As mentioned earlier, the denotation of a proper name such as *Lee* is the actual person Lee in the real world. A set of all individuals (people, animals, plants, objects, etc.) in the world, called D, stands for the domain of individuals. Then, Lee is a member of this set, as illustrated in (19):

(19) D = {Lee, Kim, Park, ...}

As previously mentioned, the denotation of a sentence is its truth-value, which is represented by the set containing the two numbers 1 for true and 0 for false.

(20) {0, 1}

Now the denotation of the VP is to be determined. Given the individual denoted by the subject NP as an argument, the VP must produce the value of the whole sentence. Here is where the functional application kicks in. The denotation of the VP must be a function from individuals such as Lee to truth-values. In our case, it denotes a function that maps those who run to 1 and all others to 0. Finally, the function is applied to a particular argument such as Lee, and we get 1 if he runs and 0 if he doesn't. Such a function is called a **characteristic function**. An example is given in the table form in (21).

(21) $f = \begin{pmatrix} \text{Lee} & \rightarrow 1 \\ \text{Kim} & \rightarrow 1 \\ \text{Park} & \rightarrow 0 \end{pmatrix}$

The full composition process is laid out in (22). Double brackets are used to represent the function that assigns a denotation to an expression: [[a]] is the