Cambridge University Press
978-1-107-01417-6 - Unification Grammars
Nissim Francez and Shuly Wintner
Excerpt
More information

1
Introduction

Natural languages[1] are among Nature's most extraordinary phenomena. While humans acquire language naturally and use it with great ease, the formalization of language, which is the focus of research in *linguistics*, remains evasive. As in other sciences, attempts at formalization involve *idealization*: ignoring exceptions, defining fragments, and the like. In the second half of the twentieth century, the field of linguistics has undergone a revolution: The themes that are studied, the vocabulary with which they are expressed, and the methods and techniques for investigating them have changed dramatically. While the traditional aims of linguistic research have been the description of particular languages (both synchronically and diachronically), sometimes with respect to other, related languages, modern theoretical linguistics seeks the *universal* principles that underlie all natural languages; it is looking for structural generalizations that hold across languages, as well as across various phrase types in a single language, and it attempts to delimit the class of possible natural languages by formal means.

The revolution in linguistics, which is attributed mainly to Noam Chomsky, has influenced the young field of *computer science*. With the onset of programming languages, research in computer science began to explore different kinds of languages: *formal* languages that are constructed as a product of concise, rigorous *rules*. The pioneering work of Chomsky provided the means for applying the results obtained in the study of natural languages to the investigation of formal languages.

One of the earliest areas of study in computer science was human cognitive processes, in particular natural languages. This area of research is

[1] Here and throughout the book we refer by 'language' to written language only, unrelated to any acoustic phenomenon associated with speech. In addition, we detach language from its actual use, in particular, in human communication.

today categorized under the term *artificial intelligence* (AI); the branch of AI that studies human languages is usually referred to as *natural language processing* (NLP). The main objective of NLP is to computationally simulate processes related to the human linguistic faculty. The ordinary instruments in such endeavors are heuristic techniques that aid in constructing practical applications. The outcomes of such research are computer systems that perform various tasks associated with language, such as question answering, text summarization, and categorization. The application to Internet search gave another boost to NLP.

A different scientific field, which lies at the crossroads of computer science and linguistics, has obtained the name *computational linguistics*. While it is related to NLP, there are distinct differences between the two. Computational linguistics studies the structure of natural languages from a formal, mathematical and computational, point of view. It is concerned with all subfields of the traditional linguistic research: phonology (the theory of speech sounds), morphology (the structure of words), syntax (the structure of sentences), semantics (the theory of meaning), and pragmatics (the study of language use and its relation to the non-linguistic world). But it approaches these fields from a unique point of departure: It describes linguistic phenomena in a way that, at least in principle, is computationally implementable.

As an example, consider the phenomenon of anaphoric reference. In virtually all natural languages, it is possible to refer, within discourse, to some entities that were mentioned earlier in the discourse through the use of pronouns. Consider, for example, the following English sentence:

> And he dreamed that there was a ladder set up on the earth, and the top of it
> reached to heaven; and the angels of God were ascending and descending on it!
> (Genesis 28:12)

The pronoun it (in both of its occurrences) refers back to the noun phrase a ladder. While most speakers of English would have no problem recognizing this fact, a formal explanation for it seems to require a substantial amount of knowledge. From a purely syntactic point of view, nothing prevents the pronoun from referring to other previously mentioned entities, such as the earth or heaven. Computational linguistic approaches to anaphora devise algorithms that relate pronouns occurring in texts with the entities to which they refer (this process is known as *anaphora resolution*). While some approaches – to this problem as well as to others – are purely analytic, others are based on probabilistic tools and the survey of online corpora of texts.

In this book we view natural languages in a formal way; more specifically, we assume that there exists a set of formal, concise rules, mathematically expressible, that characterizes the syntax of every natural language.[2] When the grammars of natural languages are formal entities, they can be naturally subjected to the application of various paradigms and techniques from computer science. The rules that govern the syntax of natural languages, and in particular, the formal way to express them, are the focus of this book.

1.1 Syntax: the structure of natural languages

It is a well-observed fact that natural languages have *structure*. Words in the sentences of any natural language are not strung together arbitrarily; rather, there are underlying rules that determine how words combine to form *phrases*, and phrases combine to yield sentences. Such rules are based on the observation that the context constrains, to a great extent, the possible words and phrases that can occur in it.

As an example, consider the sentence quoted above:

And he dreamed that there was a ladder set up on the earth

Observe that the prefix of the sentence determines its possible extension. For example, after reading And he dreamed that there was a, readers expect words from a certain category (*nouns*); whereas after And he dreamed that there was, one would expect (among other things, perhaps) phrases whose structure is different (*noun phrases*).

Exercise 1.1. What kind of phrases can possibly follow And he dreamed that? Try to characterize them as best as you can.

Syntax is the area of linguistics that assigns structure to utterances, thus determining their acceptability. It cannot, however, be viewed independently of other areas of linguistics. Syntax is an indispensable means for assigning meaning to utterances; most theories of semantics rely on syntax in that they define meanings compositionally. Thus, the meaning of a phrase is defined as a function of the meanings of its subparts. Furthermore, the principles of syntax are believed by many researchers to be responsible for the structure of words; that is, morphology is viewed by many as a subfield of syntax. Syntax also has an important influence on phonology: the structure of utterances occasionally affects phonological processes. While we do not discuss these phenomena in this book, they contribute to the importance of syntax as a central field in linguistics.

[2] We are interested in a synchronic, rather than a diachronic, description of languages: We are concerned with the features of languages at a given point in time, ignoring historic changes.

Syntax, then, is concerned with the structure of natural languages. It does so by providing the means for specifying *grammars*. A grammar is a concise description of the structure of some language. Its function is multifaceted. First and foremost, it specifies the set of sentences in the language. Second, it assigns some structure to each of these sentences. These two tasks are discussed later in Section 1.5, and in more detail in Chapter 4. Finally, it can be used to inform computational algorithms that can then analyze sentences. Such algorithms, called *parsers*, are discussed in Chapter 6. For a grammar to be used for a computational application, it must be formally defined. The next section discusses grammatical formalisms, that is, formal languages for specifying grammars.

1.2 Linguistic formalisms

Natural languages are natural phenomena, and linguistics can be viewed as part of the natural sciences: Just as physicists study the structure of the universe, so do linguists study the structure of languages. Just as physics formulates claims about the material world, linguistics is also an empirical science, formulating empirical claims about (one or more) languages that can be verified or falsified. Usually, such claims are validated by *informants*: native speakers, who pass judgments on the predictions of the theory; recently, online corpora of texts were used to achieve the same aim.[3] And just as physicists need an underlying formalism with which to express their theories (in general, mathematics), so do linguists. A clear line should be drawn between the linguistic theories that are organized, coherent sets of generalizations regarding languages, on one hand, and the formalisms in which they are expressed, on the other hand.

This book is dedicated to describing the underlying formalisms in which several contemporary linguistic theories, notably, lexical-functional grammar (LFG) and head-driven phrase structure grammar (HPSG) are expressed: unification grammars. Continuing the analogy to physics, this book should be viewed as an elementary textbook for current physics (i.e., linguistic) theories, focusing on the necessary underlying mathematics (e.g., differential equations) and its use in describing physical (linguistic) laws.

Why should there be any mathematics involved with linguistic theories? Consider what is needed from a good linguistic theory: It must be capable of describing facts about the world (in this case, facts about natural languages), but it must also be able to make *generalizations*; obviously, a comprehensive

[3] Unlike the natural sciences, linguistics "observables" are open to interpretation. For example, not all native speakers may agree on the grammaticality of some written utterance, and sometimes performance constraints can obscure competence judgments. Observing language can hardly be detached from the daily practice of its use as an actual communication means.

list of facts is impossible to come with, given the infinite nature of human languages. The language we have used thus far for describing (very roughly) the structure of sentences is English. Indeed, for many years the structure of natural languages was expressed in natural languages. The problem with using natural language to express theories is that it is informal. It is possible to give a characterization of phenomena using English, but to account for them formally, a more rigorous means is needed. Such tools are called *linguistic formalisms*. In particular, using mathematics to specify and reason about languages allows for *proofs* of claims about them.

A linguistic formalism is a (formal) language, with which claims about (natural, but also formal) languages can be made. In general, one builds a (mathematical) *model* of a (fragment of a) natural language within such a formalism. Then, within the model, claims can be *proved* (deductively), forming predictions with empirical contents that can be confronted with actual facts. Furthermore, the models, when formulated in a *computational* formalism, enable machine implementation of linguistic analyses, for example, parsers.

What is required from such a formalism? First and foremost, it should be formal; natural languages will not do. It must also be recursive, or in other words, must provide finite means for expressing infinite sets. It must be precise so that the subtleties of the theory can be easily and accurately expressed. And it must be expressive enough that the wealth of phenomena of which natural languages consist can be accounted for. Of course, the question of expressive power (the formal class of languages that can be defined with the formalism) cannot even be posed when a formalism is expressed in a natural language, rather than mathematically.

The choice of a linguistic formalism carries with it linguistic consequences: It implies that certain generalizations and predictions are more central than others. As an example, consider the notion of *verb valence*, whereby verbs are subcategorized according to the number and type of arguments they expect to have (in simple terms, the distinction between intransitive, transitive, and ditransitive verbs). Expressing such a notion in a linguistic formalism has the immediate consequence that concatenation and constituent order become dominant factors (as opposed to, say, the length of words or their distance from the beginning of the sentence). When a formalism is based on phrase-structure rules, it presupposes the notion of phrase structure (as opposed to, say, dependency-based formalization). When a formalism is highly lexicalized, it highlights the importance of the lexicon in linguistic theory, perhaps at the expense of other components of the grammar.

In this book we focus on unification-based formalisms, promoting the importance of *feature structures* and of phrase-structure rules based on feature structures for linguistic expression. Some of the consequences of this choice include reliance on phrase structure (with concatenation as the sole string-combination operation); centrality of the lexicon; very powerful and very general rules; and potentially very detailed analyses, due to the use of (deeply embedded) features and values. We advocate for the benefit of this type of formalism to linguistic theory in Chapter 5.

1.3 A gradual description of language fragments

This book deals with formalisms for describing *natural languages*. The concept of *natural* languages (as opposed to artificial ones) is very hard to define, and we do not attempt to do so here. Instead, we informally characterize small fragments of certain languages, notably English, which we use to exemplify the material explained in the book, especially in Chapter 5. We start with E_0, which is a small fragment of English; E_0 is then extended in different directions, forming broader-coverage subsets. Indeed, where such small fragments are concerned, it is rather simple to speak about similar sublanguages of other languages, and we will do so for a variety of languages, including French, Hebrew, Russian, and German. Although we do not present any formal account of the similarities among the language fragments, our intention is to account for similar phenomena. For example, when E_0 accounts for local structures, involving the expression of basic predications (a verb as a relation, and its arguments), similar phenomena in, say, Hebrew are expressed in H_0, possibly by different means.

One of the major issues discussed in this book is the quest for an adequate formal definition for natural languages. However, to account for small fragments, such as E_0, we can use an informal description. This is what we do in the material that follows.

1.3.1 Basic sentences

E_0 is a small fragment of English consisting of very simple sentences, constructed with only intransitive and transitive (but no ditransitive) verbs, common nouns, proper names, pronouns, and determiners. Typical sentences are

A sheep drinks.
Rachel herds the sheep.
Jacob loves her.

Similar strings are not E_0- (and not English-) sentences[4]:

∗Rachel feed the sheep
∗Rachel feeds herds the sheep

[4] Recall that a string preceded by '∗' is ungrammatical.

∗The shepherds feeds the sheep
∗Rachel feeds
∗Jacob loves she
∗Jacob loves Rachel she
∗Them herd the sheep

Of course, many strings are not E_0 sentences, although they are perfectly grammatical English sentences:

Rachel has seen Jacob.

However, this uses a *tense* that is outside the scope of our discussion.

Rachel was loved by Jacob

is in the *passive* voice, which we do not deal with. Other English sentences that are outside the scope of E_0 are covered by later fragments.

All E_0 sentences have two components, a *subject*, realized as a noun phrase, and a *predicate*, realized as a verb phrase. A noun phrase can be a proper name, such as Rachel, or a pronoun, such as they, or a common noun, possibly preceded by a determiner: the lamb or three sheep. A verb phrase consists of a verb, such as feed or sleeps, with a possible additional object, which is a noun phrase.

Furthermore, there are constraints on the combination of phrases in E_0. We list three of them, informally:

- The subject and the predicate must *agree* on number and person: If the subject is a third person singular, so must the verb be.
- Objects complement only – and all – the *transitive* verbs.
- When a pronoun is used, it is in the *nominative* case if it is in the subject position, and in the *accusative* case if it is an object.

As can be seen, the examples of sentences given in this section obey all the restrictions, whereas each of the nonsentences violates at least one of them. A major part of modeling this fragment involves defining the means through which these constraints can be formulated and enforced.

1.3.2 Subcategorization

In presenting linguistic examples, we extend E_0 in several directions. While the formal definitions of the language fragments that are used to illustrate a presentation are given in the appropriate places in the text, we sketch these future extensions here. Our first concern is to refine E_0 so that the *valence* of verbs is better accounted for. This means that the distinction among intransitive, transitive, and ditransitive verbs will be made more refined: E_{subcat} is a fragment of English, based on E_0, in which verbs are classified into subclasses according to the complements they "require." Recall that in E_0, transitive verbs occur with (noun phrase) objects, and intransitive verbs do not. In E_{subcat} transitive verbs can occur with different kinds of objects. For example, verbs such as eat, see, and love require a noun-phrase object; but verbs such as say and think can take a sentential object; verbs such as want, try and tend take an infinitival verb phrase; some verbs require prepositional phrases, sometimes with a specific preposition.[5] E_{subcat} will also allow verbs to have more than one object: Verbs such as give and sell can occur with two complements. For example, the following sentences are in E_{subcat}:

Laban gave Jacob his daughter.
Jacob promised Laban to marry Leah.
Laban persuaded Jacob to promise him to marry Leah.

Similar strings that violate this constraint are:

∗Rachel feeds Jacob the sheep
∗Jacob saw to marry Leah

The modeling problem here is to specify valence and enforce its saturation.

1.3.3 Control

With the addition of infinitival complements in E_{subcat}, $E_{control}$ can capture constraints of argument *control* in English. Informally, what we mean by control is the phenomenon by which certain verbs that require a sentential complement allow this complement to be incomplete: Usually, the subject of the complement is missing. The missing constituent is implicitly understood as one of the main verb's other complements: either the subject or the object. For example, the verb promise takes two objects, one of which is an infinitival verb phrase. The understood subject of this verb phrase is the subject of promise. In the sentence

Jacob promised Laban to work seven years,

it is the subject Jacob that is the understood subject of the verb work. On the other hand, in

Laban persuaded Jacob to work seven years,

it is the object Jacob that is the understood subject of work. The difference lies in the main verb: Promise is said to be a *subject control* verb, whereas persuade is an *object control* verb. In $E_{control}$, phrases that contain infinitival complements are assigned a structure that reflects their intended interpretation.

[5] We do not account for prepositional phrases in this fragment.

Here, the modeling problem is to identify the "missing" controlled part, distinguish between the two cases, and provide the means for "filling" the gap correctly.

1.3.4 Long-distance dependencies

We now consider another extension of E_{subcat}, namely E_{ldd}, typical sentences of which are

(1) The shepherd wondered whom Jacob loved ␣.
(2) The shepherd wondered whom Laban thought Jacob loved ␣.
(3) The shepherd wondered whom Laban thought Rachel claimed Jacob loved ␣.

In all these sentences (and, clearly, the sequence can be prolonged indefinitely), the transitive verb loves occurs without an explicit noun phrase in the object position. The symbol '␣', called a *gap*, is a place holder, positioned at the alleged surface location of the "missing" object. An attempt to replace the gap with an explicit noun phrase results in ungrammaticality:

(4) ∗The shepherd wondered whom Jacob loved Rachel.

However, despite the absence of the object of loves from its surface position, there is another element in the surface structure, namely whom, which is the "understood" object of loves. In some theories, it is considered a "dislocated" object (due to movement transformation). More abstractly, whom is referred to as the *filler* of the gap. It is important to notice that the sequence (1)–(3) (and its extensions) shows that there is no (theoretical, or principled) bound on the surface distance between a gap and its filler. This unboundedness motivated the use of the term *long-distance dependencies*, or *unbounded dependencies*, for such phenomena.

Some comments are needed regarding long-distance dependencies. First, note that the gap need not be in the object position. Sentences (5)–(6) show the beginning of a similar chain of sentences, in which there is a gap in the subject position of an embedded clause:

(5) Jacob wondered who ␣ loved Leah.
(6) Jacob wondered who Laban believed ␣ loved Leah.

Again, an explicit noun phrase filling the gap results in ungrammaticality:

(7) ∗Jacob wondered who the shepherd loved Leah

The filler here is the "understood" subject. Also, note that more than one gap may be present in a sentence (and, hence, more than one filler), as shown in (8a,b):

CAMBRIDGE

Cambridge University Press
978-1-107-01417-6 - Unification Grammars
Nissim Francez and Shuly Wintner
Excerpt
More information

(8a) This is the well which Jacob is likely to — draw water from —.
(8b) It was Leah that Jacob worked for — without loving —.

In some languages (e.g., Norwegian) there is no (principled) bound on the number of gaps that can occur in a single clause. Multiple gaps are outside the scope of our discussion.

There are other fragments of English in which long-distance dependencies are manifested in other forms. One example, which we do not cover here, is *topicalization*, as shown in (9)–(10):

(9) Rachel, Jacob loved —
(10) Rachel, every shepherd knew Jacob loved —

Another example is the *interrogative sentence*, such as in (11)–(12):

(11) Who did Jacob love —?
(12) Who did Laban believe Jacob loved —?

Here again, the modeling problem involves identifying the "missing" constituent and providing the means for correlating the "gap" with the dislocated pronoun that "fills" it.

1.3.5 Relative clauses

A different yet similar case of remote dependencies is the *relative clause*; these are clause that modify nouns, and typically an element is missing in the clause that is semantically "filled" by the noun being modified. Consider the following examples:

(13) The lamb that Rachel loves
(14) The lamb that loves Rachel

In (13), the noun lamb is modified by the clause that Rachel loves; note that the subcategorization constraints of *loves* are violated, since an accusative object is not explicit. Rather, the "understood" object of loves is the head noun lamb. In (14) a similar situation is exemplified, but it is the *subject* of loves, rather than its object, that is missing (and is identified with the head noun lamb).

Relative clauses can be much more complicated than these two simple examples; the "missing" element can be embedded deeply within the clause, and in some cases, it is replaced by a pronoun. We only address the simple cases in which the head noun fills the position of either the subject or the direct object of the relative clause; this fragment of English, which is an extension of E_{subcat}, is called E_{relcl}. Modeling this case requires mechanisms very similar to those used in the modeling of long-distance dependencies.