IV. LANGUAGE AS A MEANING ⇔ TEXT TRANSFORMER

IN THIS CHAPTER, we will return to linguistics, to make a review of several viewpoints on natural language, and to select one of them as the base for further studies. The components of the selected approach will be defined, i.e., *text* and *meaning*. Then some conceptual properties of the linguistic transformations will be described and an example of these transformations will be given.

POSSIBLE POINTS OF VIEW ON NATURAL LANGUAGE

One could try to define natural language in one of the following ways:

- The principal means for expressing human thoughts;
- The principal means for text generation;
- The principal means of human communication.

The first definition—"the principal means for expressing human thoughts"—touches upon the expressive function of language. Indeed, some features of the outer world are reflected in the human brain and are evidently processed by it, and this processing is just the human thought. However, we do not have any real evidence that human beings directly use words of a specific natural language in the process of thinking. Modes of thinking other than linguistic ones are also known. For example, mathematicians with different native languages can have the same ideas about an abstract subject, though they express these thoughts in quite different words. In addition, there are kinds of human thoughts—like operations with musical or visual images—that cannot be directly reduced to words.

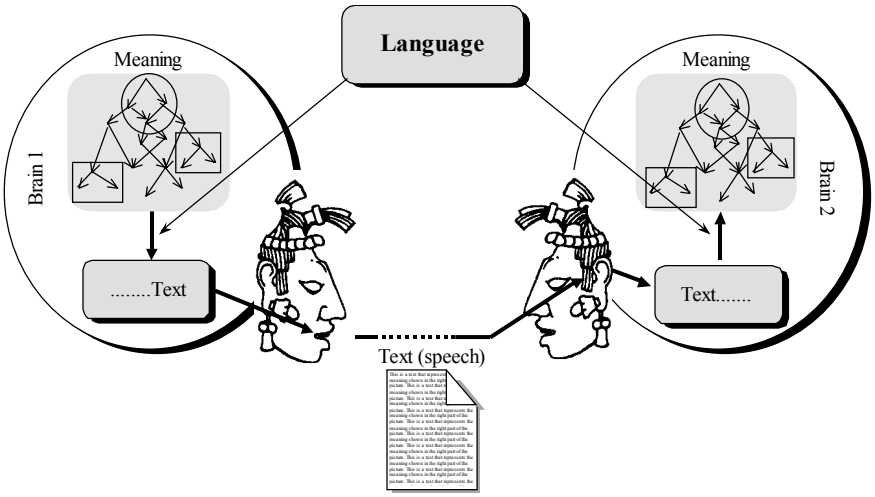

FIGURE IV.1. *The role of language in human communication.*

As to the second definition—"the principal means for text generation"—there is no doubt that the flow of utterances, or texts, is a very important result of functioning of natural language.

However, communication includes not only generation (speaking), but also understanding of utterances. This definition also ignores the starting point of text generation, which is probably the target point of understanding. Generation cannot exist without this starting point, which is contents of the target utterance or the text in the whole. We can call these contents *meaning*.

As to the third definition—"the principal means of human communication,"—we can readily agree that the communicative function is the main function of natural language.

Clearly, only persons living in contact with society really need a language for efficient communication. This definition is perhaps correct, but it does not touch upon two main aspects of communication in natural language, namely, speaking and understanding, and thus it does not try to define these aspects, separately and in their interaction.

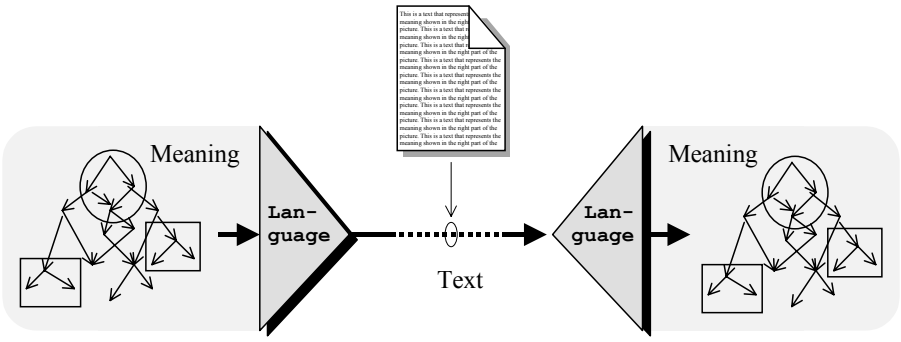

FIGURE IV.2. *Language functions like encoder / decoder in a communication channel.*

Thus, these definitions are not sufficient for our purposes. A better definition should touch upon all useful components of the ones given above, such as *text*, *meaning*, *generation*, and *understanding*. Such definition will be given in the next section.

LANGUAGE AS A BI-DIRECTIONAL TRANSFORMER

The main purpose of human communication is transferring some information—let us call it Meaning[6]—from one person to the other. However, the direct transferring of thoughts is not possible.

Thus, people have to use some special physical representation of their thoughts, let us call it Text.[7] Then, language is a tool to transform one of these representations to another, i.e. to transform Meanings to words when speaking, and the words to their Meaning when listening (see Figure IV.1).

[6] We use capitalization to distinguish the terms Meaning and Text in their specific sense used in the Meaning ⇔ Text Theory from the conventional usage of these words.

[7] For the sake of the argument, here we consider speech, as well as written text, to be a kind of Text.

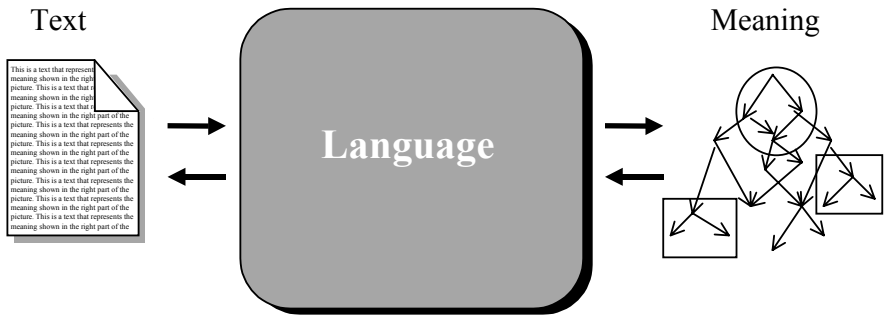

FIGURE IV.3. *Language as a Meaning ⇔ Text transformer.*

It is important to realize that the communicating persons use the same language, which is their common knowledge, and each of them has a copy of it in the brain.

If we compare this situation with transferring the information over a communication channel, such as a computer network, the role of language is encoding the information at the transmitting end and then decoding it at the receiving end.[8] Again, here we deal with two copies of the same encoder/decoder (see Figure IV.2).

Thus, we naturally came to the definition of natural language as a transformer of Meanings to Texts, and, in the opposite direction, from Texts to Meanings (see Figure IV.3).

This transformer is supposed to reside in human brain. By transformation we mean some form of translation, so that both the Text and the corresponding Meaning contain the same information. What we specifically mean by these two concepts, Text and Meaning, will be discussed in detail later.

Being originally expressed in an explicit form by Igor Mel'čuk, this definition is shared nowadays by many other linguists. It permits to recognize how computer programs can simulate, or model, the capacity of the human brain to transform the information from one of these representations into another.

[8] It is not surprising that later in this book, levels of information representation will appear, in direct analogy with the modern computer network protocols.

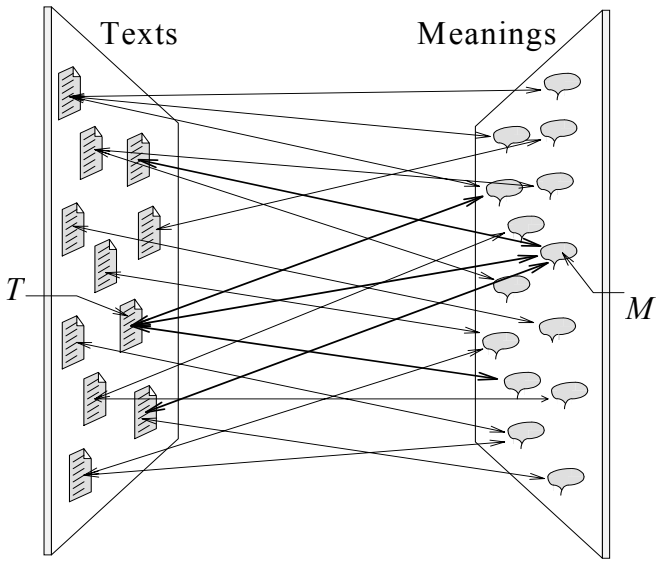

FIGURE IV.4. *Meaning ⇔ Text many-to-many mapping.*

Essentially, this definition combines the second and the third definitions considered in the previous section. Clearly, the transformation of Text into Meaning and vice versa is obligatory for any human communication, since it implies transferring the Meaning from one person to another using the Text as its intermediate representation. The transformation of Meaning into Text is obligatory for the generation of utterances. To be more precise, in the whole process of communication of human thoughts the definition 1 given earlier actually refers to Meaning, the definition 2 to Text, and the definition 3 to both mentioned aspects of language.

With our present definition, language can be considered analogous to a technical device, which has input and output. Some information, namely Text, being entered to its input, is transformed into another form with equivalent contents.

The new form at the output is Meaning. More precisely, we consider a bi-directional transformer, i.e., two transformers working in

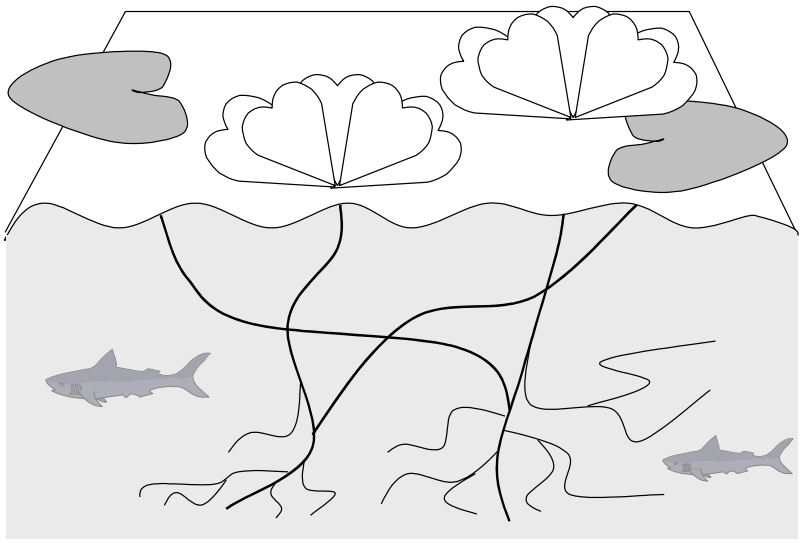

FIGURE IV.5. *Metaphor of surface and deep structures.*

parallel but in opposite directions. Text is the result of the activity of one of these transformers, and Meaning, of the other.

Programmers can compare such a device with a compiler, let us say, a C++ compiler, which takes a character file with the ASCII text of the program in the input and produces some binary code with machine instructions, as the output. The binary code corresponds to the meaning of the program. However, a compiler usually cannot translate the machine instructions back to a C++ program text.

As a mathematical analogy to this definition, we can imagine a bi-directional mapping between one huge set, the set of all possible Texts, and another huge set, the set of all possible Meanings (see Figure IV.4).

The two sets, Texts and Meanings, are not quite symmetric in their properties. Only the Texts have an explicit expression, only they can be immediately observed or directly transferred from one person to another, while the Meanings reside in the brain of each person independently and cannot be immediately observed or assessed.

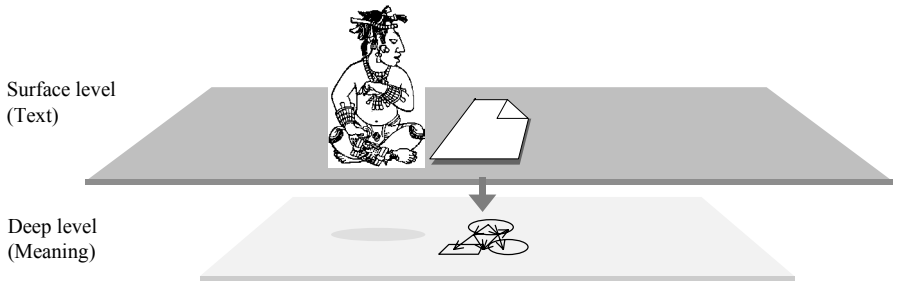

FIGURE IV.6. *Two levels of representation.*

This is similar to the front panel of an electrical device: the lights and switches on its *surface* can be observed, while the electrical processes represented by the lights and controlled by the switches are *deep*[9] under the cover of the device, and can only be guessed by watching the lights or experimenting with the switches.

Another metaphor of surface and deep structures of language is shown in Figure IV.5. We can directly observe the surface of water, but in order to learn what leaf is connected to what flower through common roots, we need to analyze what is under the surface. There is much more below the surface than on top of it, and only analysis of the deeper phenomena gives us understanding of the whole thing.

All this is often considered as surface and deep *levels of the representation* of utterances (see Figure IV.6). The man on the picture cannot see the meaning of the text immediately and has to penetrate below the surface of the text to find its meaning.

Thus, the set of Texts is considered the *surface* edge of the Meaning ⇔ Text transformer, while the set of Meanings gives its *deep* edge. The Meaning corresponding to the given Text at the depth is also called its *semantic representation*.

The transformation of Meaning into Text is called *synthesis* of the Text. The transformation to the inverse direction, that is from Text

[9] See also discussion of the terms *surface* and *deep* in comparison with their usage in generative grammars on the page 124.

into Meaning, is called *analysis* of Text. Thus, according to our basic definition, natural language is both *analyzer* and *synthesizer* of Texts, at the same time.

This definition uses the notions of Text and Meaning, although they have been neither defined nor described so far. Such descriptions will be given in the following sections.

TEXT, WHAT IS IT?

The empirical reality for theoretical linguistics comprises, in the first place, the sounds of speech. Samples of speech, i.e., separate words, utterances, discourses, etc., are given to the researchers directly and, for living languages, are available in an unlimited supply.

Speech is a continuous flow of acoustic signals, just like music or noise. However, linguistics is mainly oriented to the processing of natural language in a discrete form.

The discrete form of speech supposes dividing the flow of the acoustic signals into sequentially arranged entities belonging to a finite set of partial signals. The finite set of all possible partial signals for a given language is similar to a usual alphabet, and is actually called a *phonetic alphabet*.

For representation of the sound of speech on paper, a special *phonetic transcription* using *phonetic symbols* to represent speech sounds was invented by scientists. It is used in dictionaries, to explain the pronunciation of foreign words, and in theoretical linguistics.

A different, much more important issue for modern computational linguistics form of speech representation arose spontaneously in the human practice as the written form of speech, or the writing system.

People use three main writing systems: that of alphabetic type, of syllabic type, and of hieroglyphic type. The majority of humankind use alphabetic writing, which tries to reach correspondence between letters and sounds of speech.

Two major countries, China and Japan,[10] use the hieroglyphic writing. Several countries use syllabic writing, among them Korea. *Hieroglyphs* represent the meaning of words or their parts. At least, they originally were intended to represent directly the meaning, though the direct relationship between a hieroglyph and the meaning of the word in some cases was lost long ago.

Letters are to some degree similar to sounds in their functions. In their origin, letters were intended to directly represent sounds, so that a text written in letters is some kind of representation of the corresponding sounds of speech. Nevertheless, the simple relationship between letters and sounds in many languages was also lost. In Spanish, however, this relationship is much more straightforward than, let us say, in English or French.

Syllabic signs are similar to letters, but each of them represents a whole syllable, i.e., a group of one or several *consonants* and a *vowel*. Thus, such a writing system contains a greater number of signs and sometimes is less flexible in representing new words, especially foreign ones. Indeed, foreign languages can contain specific combinations of sounds, which cannot be represented by the given set of syllables. The syllabic signs usually have more sophisticated shape than in letter type writing, resembling hieroglyphs to some degree.

In more developed writing systems of a similar type, the signs (called in this case *glyphs*) can represent either single sounds or larger parts of words such as syllables, groups of syllables, or entire words. An example of such a writing system is Mayan writing (see Figure I.2). In spite of their unusual appearance, Mayan glyphs are more syllabic signs than hieroglyphs, and they usually represent the sounds of the speech rather than the meaning of words. The reader can become familiar with Mayan glyphs through the Internet site [52].

[10] In fact, Japanese language uses a mixture of hieroglyphic and syllabic symbols, though the use of syllabic symbols is limited.

Currently, most of the practical tasks of computational linguistics are connected with written texts stored on computer media. Among written texts, those written in alphabetic symbols are more usual for computational linguistics than the phonetic transcription of speech.[11] Hence, in this book the methods of language processing will usually be applied to the written form of natural language.

For the given reason, Texts mentioned in the definition of language should then be thought of as common texts in their usual written form. Written texts are chains of letters, usually subdivided into separate words by spaces[12] and punctuation marks. The combinations of words can constitute sentences, paragraphs, and discourses. For computational linguistics, all of them are examples of Texts.[13]

Words are not utmost elementary units of language. Fragments of texts, which are smaller than words and, at the same time, have their own meanings, are called *morphs*. We will define morphs more precisely later. Now it is sufficient for us to understand that a morph can contain an arbitrary number of letters (or now and then no letters at all!), and can cover a whole word or some part of it. Therefore, Meanings can correspond to some specially defined parts of words, whole words, phrases, sentences, paragraphs, and discourses.

It is helpful to compare the linear structure of text with the flow of musical sounds. The mouth as the organ of speech has rather limited abilities. It can utter only one sound at a time, and the flow of these sounds can be additionally modulated only in a very restricted manner, e.g., by stress, intonation, etc. On the contrary, a set of musical instruments can produce several sounds synchronously, form-

[11] This does not mean that the discussed methods are not applicable to phonetic transcription or, on the other hand, to hieroglyphs. However, just for simplification we will choose only the representation by letters.

[12] In some writing systems, like Japanese, words are not separated by spaces in the written text. Of course, this does not mean that these languages do not have words, but the word boundaries are not reflected in writing. As opposed to Japanese, Vietnamese separates all the syllables.

[13] In Western tradition including HPSG, Text in the given sense is called *list of phoneme strings*, or simply phonetic representation of a linguistic sign.

ing harmonies or several melodies going in parallel. This parallelism can be considered as nonlinear structuring. The human had to be satisfied with the instrument of speech given to him by nature. This is why we use while speaking a linear and rather slow method of acoustic coding of the information we want to communicate to somebody else.

The main features of a Text can be summarized as follows:

- *Meaning*. Not any sequence of letters can be considered a text. A text is intended to encode some information relevant for human beings. The existing connection between texts and meanings is the reason for processing natural language texts.

- *Linear structure*. While the information contained in the text can have a very complicated structure, with many relationships between its elements, the text itself has always one-dimensional, linear nature, given letter by letter. Of course, the fact that lines are organized in a square book page does not matter: it is equivalent to just one very long line, wrapped to fit in the pages. Therefore, a text represents non-linear information transformed into a linear form. What is more, the human cannot represent in usual texts even the restricted non-linear elements of spoken language, namely, intonation and logical stress. Punctuation marks only give a feeble approximation to these non-linear elements.

- *Nested structure* and *coherence*. A text consists of elementary pieces having their own, usually rather elementary, meaning. They are organized in larger structures, such as words, which in turn have their own meaning. This meaning is determined by the meaning of each one of their components, though not always in a straightforward way. These structures are organized in even larger structures like sentences, etc. The sentences, paragraphs, etc., constitute what is called *discourse*, the main property of which is its *connectivity*, or *coherence*: it tells some consistent story about objects, persons, or relations, common to all its parts. Such organization provides linguistics with the means to develop the methods of intelligent text processing.

Thus, we could say that linguistics studies *human* ways of *linear* encoding[14] of *non-linear* information.

MEANING, WHAT IS IT?

Meanings, in contrast to texts, cannot be observed directly. As we mentioned above, we consider the Meaning to be the structures in the human brain which people experience as ideas and thoughts. Since we do not know and cannot precisely represent those brain processes, for practical purposes we must use a representation of Meaning, which is more suitable for manipulation in a computer. Thus, for our purposes, Meaning will be identified with that representation.

In the future, neurophysiological research will eventually discover what signals really correspond to meanings in our brain, and what structure these signals have, but for now those signals remain only vaguely understood. Hence we take the pragmatic viewpoint that, if a representation we use allows a computer to manipulate and respond to texts with an ability close to that of a human, then this representation is rather good for the real Meaning and fits our purposes.

As it was described earlier, the task of language is to transform information from one representation, the Text, into another, the Meaning, and vice versa. A computer program that models the function of language must perform the same task. In any application, there should be some other system or device that consumes the results of the transformation of texts, and produces the information that is to be transformed into a text. The operations of such a device or a system is beyond the scope of computational linguistics itself. Rather, such a system *uses* the linguistic module as its interface with the outer world (see Figure IV.7).

[14] And also compression, which increases the non-linearity of the underlying structure.

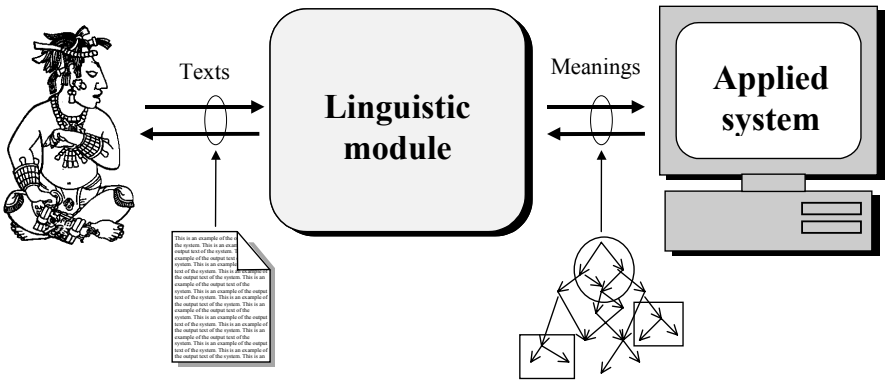

FIGURE IV.7. *Structure of an application system with a natural language interface.*

For a linguistic module of a system, the Meaning is a formal language or a format of information representation immediately understandable for, or executable on, the consumer of the information: the underlying expert or reasoning system, database, robot control system, etc. It is supposed that this underlying system produces its responses just in the same format. Thus, in practice, the format of Meaning is already given to the developers of the linguistic module for any specific application system.

Usually, such systems are aware on the *entities* mentioned in the text, their *states, properties*, *processes*, *actions*, and *relationships*.

Besides, there exists other type of information in a text, such as *beliefs, estimations,* and *intentions* of its author. For example, in the Spanish sentence ***Creo que*** *su esposa está aquí,* the part reflecting the author's belief is given in bold face. The author usually flavors any text with these elements. Words reflecting basic information, through some stylistic coloring, can additionally express author's attitude. For example, the Spanish sentence *Este* ***hombrón*** *no está trabajando ahora* has the meaning 'this man is not working now and I consider him big and coarse'.

Perhaps only very formal texts like legal documents do not contain subjective attitude of the author(s) to the reflected issues. The

advanced application system should distinguish the basic information delivered in texts from author's beliefs, estimations, and intentions.

Additionally, even a very formal text contains many explicit references and explanations of links between parts of a text, and these elements serve as a content table or a protocol of the author's information about text structuring. This is not the information about the relevant matters as such. Instead, this is some kind of meta-information about how the parts of the text are combined together, i.e., a "text about text." We will not discuss such insertions in this book. Since properties, processes, and actions can be represented as relationships between entities touched upon in Text, just these features are used to represent Meaning.

The entities, states, properties, processes, and actions are usually denoted by some names in semantic representation. These names can be compared with the names of variables and functions in a computer program. Such names have no parts of speech, so that a process or an action of, say, 'development' can be equally called in Spanish *desarrollar* or *desarrollo*, while the property of 'small' can be equally called *pequeño* or *ser pequeño*. Usually only one word-form is used for the name, so that the property itself is called, for instance, *pequeño* (neither *pequeña* nor *pequeñas*). Concerning the value *plural* of grammatical category of number, on the semantic level it is transformed to the notion of multiplicity and is represented by a separate element.

TWO WAYS TO REPRESENT MEANING

To represent the entities and relationships mentioned in the texts, the following two logically and mathematically equivalent formalisms are used:

- *Predicative formulas*. *Logical predicates* are introduced in mathematical logic. In linguistics, they are used in conventional logical notation and can have one or more arguments. Coherence

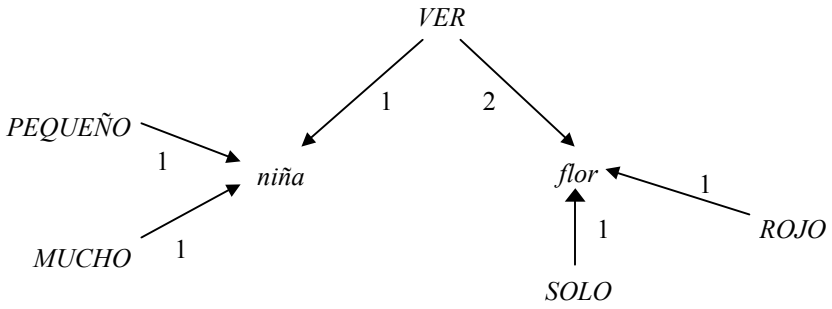

FIGURE IV.8. *Semantic network for the sentence*
Las niñas pequeñas ven la flor roja.

of a text is expressed in the first place by means of common arguments for the predicates involved. For example, the meaning of the sentence *Las niñas pequeñas ven la flor roja* is represented by the following conjunctive predicative formula:

VER(*niña, flor*) &
MUCHO(*niña*) &
PEQUEÑO(*niña*) &
SOLO(*flor*) &
ROJO(*flor*)

In such representation, predicates *SOLO* and *MUCHO* have the meanings 'number of the entities given by the argument is one' and 'number of the entities given by the argument is more than one,' respectively. Arguments of the predicates that are not predicates by themselves are called *terms*. They are written in lowercase letters, in contrast to the predicates that are written in uppercase.

- *Directed labeled graphs.* The nodes of these graphs represent the terms or predicates, and the arrows connect predicates with their arguments, i.e., terms or other predicates. The arrows are marked with numeric labels according to the number of the corresponding argument (1st argument, 2nd, etc.). Though each predicate as-

signs to each its argument a specific *semantic role*, the numerical labels in the graph are used only to distinguish the arguments. For predicates denoting actions, the label 1 usually marks the agent, the label 2, patient or target, etc. Nevertheless, a label of such type does not make any semantic allusion, the enumeration being rather arbitrary. Thus, the graph representation is just equivalent to the predicate one rather than provides any additional information. The semantic representation in the form of directed labeled graph is often called *semantic network*. Figure IV.8 shows the semantic network representation for the example above.

These two representations are equivalent, and either of them can be used. Indeed, there exist a number of easy ways to encode any network linearly.

For human readers, in books and articles, the graph representation is especially convenient. For internal structures of a computer program, the equivalent predicative representation is usually preferred, with special formal means to enumerate the referentially common arguments of various logical predicates.

The graph representation explicitly shows the commonality of the arguments, making it obvious what is known about a specific entity. In the drawing shown in the Figure IV.8, for example, it is immediately seen that there are three pieces of information about the terms *niña* and *flor*, two pieces about the predicate *VER*, and one for predicate *ROJO* and *SOLO*.

Some scientists identify the representations of Meaning with the representation of human knowledge in general[15]. The human knowledge is of concern not only for natural language processing, but also for the task of transferring knowledge between computers, whether that knowledge is expressed by natural language or by some other means. For the transfer of knowledge, it is important to standardize

[15] Within the computer community, efforts are under way to develop knowledge representation both in a linear format (KIF = Knowledge Interchange Format), and in a graphical format (CG = Conceptual Graphs).

methods of representation, so that the knowledge can be communicated accurately. Just for these purposes, the computer community is developing knowledge representation in both the linear and the graphical format. The accuracy and utility of any representation of knowledge should be verified in practice, i.e., in applications.

In the opinion of other scientists, the representations of Meaning and of human knowledge can operate by the same logical structures, but the knowledge in general is in no way coincident with purely linguistic knowledge and even can have different, non-discrete, nature. Hence, they argue, for the transition from Meaning in its linguistic sense to the corresponding representation in terms of general human knowledge, some special stage is needed. This stage is not a part of language and can operate with tools not included in those of the language proper.

DECOMPOSITION AND ATOMIZATION OF MEANING

Semantic representation in many cases turns out to be *universal*, i.e., common to different natural languages. Purely grammatical features of different languages are not usually reflected in this representation. For example, the gender of Spanish nouns and adjectives is not included in their semantic representation, so that this representation turned to be equal to that of English. If the given noun refers to a person of a specific sex, the latter is reflected on semantic level explicitly, via a special predicate of sex, and it is on the grammar of specific language where is established the correspondence between sex and gender. It is curious that in German nouns can have three genders: masculine, feminine, and neuter, but the noun *Mädchen* ‘girl’ is neuter, not feminine!

Thus, the semantic representation of the English sentence *The little girls see the red flower* it is the same as the one given above, despite the absence of gender in English nouns and adjectives. The representation of the corresponding Russian sentence is the same too, though the word used for *red* in Russian has masculine gender,

because of its agreement in gender with corresponding noun of masculine.[16]

Nevertheless, the cases when semantic representations for two or more utterances with seemingly the same meaning do occur. In such situations, linguists hope to find a universal representation via decomposition and even atomization of the meaning of several semantic components.

In natural sciences, such as physics, researchers usually try to divide all the entities under consideration into the simplest possible, i.e., atomic, or elementary, units and then to deduce properties of their conglomerations from the properties of these elementary entities. In principle, linguistics has the same objective. It tries to find the atomic elements of meaning usually called *semantic primitives*, or *semes*.

Semes are considered indefinable, since they cannot be interpreted in terms of any other linguistic meanings. Nevertheless, they can be explained to human readers by examples from the extralinguistic reality, such as pictures, sound records, videos, etc. All other components of semantic representation should be then expressed through the semes.

In other words, each predicate or its terms can be usually represented in the semantic representation of text in a more detailed manner, such as a logical formula or a semantic graph. For example, we can decompose

$$MATAR(x) \rightarrow CAUSAR(MORIR(x)) \rightarrow CAUSAR(CESAR(VIVIR(x))),$$

i.e., $MATAR(x)$ is something like '*causar cesar el vivir*(x),' or '*cause stop living*(x),' where the predicates $CESAR(x)$, $VIVIR(y)$, and $CAUSAR(z)$ are more elementary than the initial predicate $MATAR(x)$.[17]

[16] In Russian: *Malen'kie devochki vidjat krasnyj cvetok*; the two last words are singular masculine.

[17] Some researchers consider the predicate *TO LIVE*(x) elementary.

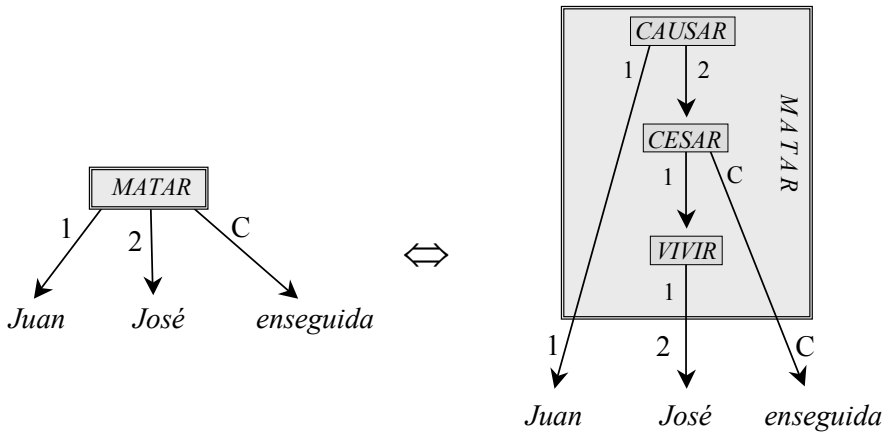

FIGURE IV.9. *Decomposition of the verb* MATAR *into semes.*

Figure IV.9 shows a decomposition of the sentence *Juan mató a José enseguida = Juan causó a José cesar vivir enseguida* in the mentioned more primitive notions. Note that the number labels of valencies of the whole combination of the primitives can differ from the number labels of corresponding valencies of the member primitives: e.g., the actant 2 of the whole combination is the actant 1 of the component *VIVIR*. The mark C in Figure IV.9 stands for the circumstantial relation (which is not a valency but something inverse, i.e., a passive semantic valency).

Over the past 30 years, ambitious attempts to find and describe a limited number of semes, to which a major part of the semantics of a natural language would be reduced, have not been successful.

Some scientists agree that the expected number of such semes is not much more than 2´000, but until now, this figure is still debatable. To comply with needs of computational linguistics, everybody agreed that it is sufficient to disintegrate meanings of lexemes to a reasonable limit implied by the application.

Therefore, computational linguistics uses many evidently non-elementary terms and logical predicates in the semantic representation. From this point of view, the translation from one cognate language to another does not need any disintegration of meaning at all.

Once again, only practical results help computational linguists to judge what meaning representation is the best for the selected application domain.

NOT-UNIQUENESS OF MEANING ⇒ TEXT MAPPING: SYNONYMY

Returning to the mapping of Meanings to Texts and vice versa, we should mention that, in contrast to common mathematical functions, this mapping is not unique in both directions, i.e., it is of the many-to-many type. In this section, we will discuss one direction of the mapping: from Meanings to Texts.

Different texts or their fragments can be, in the opinion of all or the majority of people, equivalent in their meanings. In other words, two or more texts can be mapped to the same element of the set of Meanings. In Figure IV.4, the Meaning *M* is represented with three different Texts *T*, i.e., these three Texts have the same Meaning.[18]

For example, the Spanish adjectives *pequeño* and *chico* are equivalent in many contexts, as well as the English words *small* and *little*. Such equivalent words are called *synonymous words*, or *synonyms*, and the phenomenon is called *synonymy* of words. We can consider also synonymy of word combinations (phrases) or sentences as well. In these cases the term *synonymous expressions* is used.

The words equivalent in all possible contexts are called *absolute synonyms*. Trivial examples of absolute synonymy are abbreviated and complete names of organizations, e.g. in Spanish *ONU* ≡ *Organización de las Naciones Unidas*. Nontrivial examples of absolute synonymy of single words are rather rare in any language. Examples from Mexican Spanish are: *alzadura* ≡ *alzamiento*, *acotación* ≡ *acotamiento*, *coche* ≡ *carro*.

However, it is more frequent that the two synonyms are equivalent in their meanings in many contexts, but not all.

[18] This is shown by the three bold arrows in Figure IV.4.

Sometimes the set of possible contexts for one such synonym covers the whole set of contexts for another synonym; this is called *inclusive synonymy*. Spanish examples are *querer* > *desear* > *anhelar*: *querer* is less specific than *desear* which in turn is less specific than *anhelar*. It means that in nearly every context we can substitute *desear* or *querer* for *anhelar*, but not in every context *anhelar* can be substituted for *querer* or *desear*.

Most frequently, though, we can find only some—perhaps significant—intersection of the possible sets of contexts. For example, the Spanish nouns *deseo* and *voluntad* are exchangeable in many cases, but in some cases only one of them can be used.

Such *partial synonyms* never have quite the same meaning. In some contexts, the difference is not relevant, so that they both can be used, whereas in other contexts the difference does not permit to replace one partial synonym with the other.

The book [24] is a typical dictionary of synonyms in printed form. The menu item *Language* | *Synonyms* in Microsoft Word is a typical example of an electronic dictionary of synonyms. However, many of the words that it contains in partial lists are not really synonyms, but related words, or partial synonyms, with a rather small intersection of common contexts.

NOT-UNIQUENESS OF TEXT ⇒ MEANING MAPPING: HOMONYMY

In the opposite direction—Texts to Meanings—a text or its fragment can exhibit two or more different meanings. That is, one element of the surface edge of the mapping (i.e. text) can correspond to two or more elements of the deep edge. We have already discussed this phenomenon in the section on automatic translation, where the example of Spanish word *gato* was given (see page 72). Many such examples can be found in any Spanish-English dictionary. A few more examples from Spanish are given below.

- The Spanish adjective *real* has two quite different meanings corresponding to the English *real* and *royal*.

- The Spanish verb *querer* has three different meanings corresponding to English *to wish*, *to like*, and *to love*.
- The Spanish noun *antigüedad* has three different meanings:
  - 'antiquity', i.e. a thing belonging to an ancient epoch,
  - 'antique', i.e. a memorial of classic antiquity,
  - 'seniority', i.e. length of period of service in years.

The words with the same textual representation but different meanings are called *homonymous* words, or *homonyms,* with respect to each other, and the phenomenon itself is called *homonymy*. Larger fragments of texts—such as word combinations (phrases) or sentences—can also be homonymous. Then the term *homonymous expressions* is used.

To explain the phenomenon of homonymy in more detail, we should resort again to the strict terms *lexeme* and *wordform*, rather than to the vague term *word*. Then we can distinguish the following important cases of homonymy:

- *Lexico-morphologic homonymy*: two wordforms belong to two different lexemes. This is the most general case of homonymy. For example, the string *aviso* is the wordform of both the verb *AVISAR* and the noun *AVISO*. The wordform *clasificación* belong to both the lexeme *CLASIFICACIÓN*$_1$ 'process of classification' and the lexeme *CLASIFICACIÓN*$_2$ 'result of classification,' though the wordform *clasificaciones* belongs only to *CLASIFICACIÓN*$_2$, since *CLASIFICACIÓN*$_1$ does not have the plural form. It should be noted that it is not relevant whether the name of the lexeme coincides with the specific homonymous wordform or not.

  Another case of lexico-morphologic homonymy is represented by two different lexemes whose sets of wordforms intersect in more than one wordforms. For example, the lexemes *RODAR* and *RUEDA* cover two homonymous wordforms, *rueda* and *ruedas*; the lexemes *IR* and *SER* have a number of wordforms in common: *fui*, *fuiste*, ..., *fueron*.

- Purely *lexical homonymy*: two or more lexemes have the same sets of wordforms, like Spanish *REAL*$_1$ 'real' and *REAL*$_2$ 'royal' (the both have the same wordform set {*real*, *reales*}) or *QUERER*$_1$ 'to wish,' *QUERER*$_2$ 'to like,' and *QUERER*$_3$ 'to love.'
- *Morpho-syntactic homonymy*: the whole sets of wordforms are the same for two or more lexemes, but these lexemes differ in meaning and in one or more morpho-syntactic properties. For example, Spanish lexemes (*el*) *frente* 'front' and (*la*) *frente* 'forehead' differ, in addition to meaning, in gender, which influences syntactical properties of the lexemes.
- Purely *morphologic homonymy*: two or more wordforms are different members of the wordform set for the same lexeme. For example, *fáciles* is the wordform for both masculine plural and feminine plural of the Spanish adjective *FÁCIL* 'easy.' We should admit this type of homonymy, since wordforms of Spanish adjectives generally differ in gender (e.g., *nuevos*, *nuevas* 'new').

Resolution of all these kinds of homonymy is performed by the human listener or reader according to the context of the wordform or based on the extralinguistic situation in which this form is used. In general, the reader or listener does not even take notice of any ambiguity. The corresponding mental operations are immediate and very effective. However, resolution of such ambiguity by computer requires sophisticated methods.

In common opinion, the resolution of homonymy (and ambiguity in general) is one of *the most difficult problems* of computational linguistics and must be dealt with as an essential and integral part of the language-understanding process.

Without automatic homonymy resolution, all the attempts to automatically "understand" natural language will be highly error-prone and have rather limited utility.

MORE ON HOMONYMY

In the field of computational linguistics, homonymous lexemes usually form separate entries in dictionaries. Linguistic analyzers must resolve the homonymy automatically, by choosing the correct option among those described in the dictionary.

For formal distinguishing of homonyms, their description in conventional dictionaries is usually divided into several subentries. The names of lexical homonyms are supplied with the indices (numbers) attached to the words in their standard dictionary form, just as we do it in this book. Of course, in text generation, when the program compiles a text containing such words, the indices are eliminated.

The purely lexical homonymy is maybe the most difficult to resolve since at the morphologic stage of text processing it is impossible to determine what homonym is true in this context. Since morphologic considerations are useless, it is necessary to process the hypotheses about several homonyms in parallel.

Concerning similarity of meaning of different lexical homonyms, various situations can be observed in any language. In some cases, such homonyms have no elements of meaning in common at all, like the Spanish $REAL_1$ 'real' and $REAL_2$ 'royal.' In other cases, the intersection of meaning is obvious, like in $QUERER_2$ 'to like' and $QUERER_3$ 'to love,' or $CLASIFICACIÓN_1$ 'process of classification' and $CLASIFICACIÓN_2$ 'result of classification.' In the latter cases, the relation can be exposed through the decomposition of meanings of the homonyms lexemes. The cases in which meanings intersect are referred to in general linguistics as *polysemy*.

For theoretical purposes, we can refer the whole set of homonymous lexemes connected in their meaning as *vocable*. For example, we may introduce the vocable $\{QUERER_1, QUERER_2, QUERER_3\}$. Or else we can take united lexeme, which is called *polysemic* one.

In computational linguistics, the intricate semantic structures of various lexemes are usually ignored. Thus, similarity in meaning is ignored too.

Nevertheless, for purely technical purposes, sets of any homonymous lexemes, no matter whether they are connected in meaning or not, can be considered. They might be referred as *pseudo-vocables*. For example, the pseudo-vocable $REAL = \{REAL_1, REAL_2\}$ can be introduced.

A more versatile approach to handle polysemy in computational linguistics has been developed in recent years using object-oriented ideas. Polysemic lexemes are represented as one superclass that reflects the common part of their meaning, and a number of subclasses then reflect their semantic differences.

A serious complication for computational linguistics is that new senses of old words are constantly being created in natural language. The older words are used in new meanings, for new situations or in new contexts. It has been observed that natural language has the property of self-enrichment and thus is very *productive*.

The ways of the enrichment of language are rather numerous, and the main of them are the following:

- A former lexeme is used in a *metaphorical* way. For example, numerous nouns denoting a process are used in many languages to refer also to a result of this process (*cf.* Spanish *declaración, publicación, interpretación*, etc.). The semantic proximity is thus exploited. For another example, the Spanish word *estética* 'esthetics' rather recently has acquired the meaning of heir-dressing saloon in Mexico. Since highly professional heir dressing really achieves esthetic goals, the semantic proximity is also evident here. The problem of resolution of metaphorical homonymy has been a topic of much research [51].

- A former lexeme is used in a *metonymical* way. Some proximity in place, form, configuration, function, or situation is used for metonymy. As the first example, the Spanish words *lentes* 'lenses,' *espejuelos* 'glasses,' and *gafas* 'hooks' are used in the meaning 'spectacles.' Thus, a part of a thing gives the name to the whole thing. As the second example, in many languages the name of an organization with a stable residence can be used to

designate its seat. For another example, *Ha llegado a la universidad* means that the person arrived at the building or the campus of the university. As the third example, the Spanish word *pluma* 'feather' is used also as 'pen.' As not back ago as in the middle of ninth century, feathers were used for writing, and then the newly invented tool for writing had kept by several languages as the name of its functional predecessor.

- A new lexeme is loaned from a foreign language. Meantime, the former, "native," lexeme can remain in the language, with essentially the same meaning. For example, English had adopted the Russian word *sputnik* in 1957, but the term *artificial satellite* is used as before.

- Commonly used abbreviations became common words, loosing their marking by uppercase letters. For example, the Spanish words *sida* and *ovni* are used now more frequently, then their synonymous counterparts *síndrome de inmunodeficiencia adquirida* and *objeto volante no identificado*.

One can see that metaphors, metonymies, loans, and former abbreviations broaden both homonymy and synonymy of language.

Returning to the description of all possible senses of homonymous words, we should admit that this problem does not have an agreed solution in lexicography. This can be proved by comparison of any two large dictionaries. Below, given are two entries with the same Spanish lexeme *estante* 'rack/bookcase/shelf,' one taken from the Dictionary of Anaya group [22] and the other from the Dictionary of Royal Academy of Spain (DRAE) [23].

estante (in Anaya Dictionary)

1. *m.* Armario sin puertas y con baldas.
2. *m.* Balda, anaquel.
3. *m.* Cada uno de los pies que sostienen la armadura de algunas máquinas.
4. *adj.* Parado, inmóvil.

estante (in DRAE)

1. *a. p. us*. de *estar*. Que está presente o permanente en un lugar. *Pedro, ESTANTE en la corte romana*.
2. *adj*. Aplícase al ganado, en especial lanar, que pasta constantemente dentro del término jurisdiccional en que está amillarado.
3. Dícese del ganadero o dueño de este ganado.
4. Mueble con anaqueles o entrepaños, y generalmente sin puertas, que sirve para colocar libros, papeles u otras cosas.
5. Anaquel.
6. Cada uno de los cuatro pies derechos que sostienen la armadura del batán, en que juegan los mazos.
7. Cada uno de los dos pies derechos sobre que se apoya y gira el eje horizontal de un torno.
8. *Murc*. El que en compañía de otros lleva los pasos en las procesiones de Semana Santa.
9. *Amér*. Cada uno de los maderos incorruptibles que, hincados en el suelo, sirven de sostén al armazón de las casas en las ciudades tropicales.
10. *Mar*. Palo o madero que se ponía sobre las mesas de guarnición para atar en él los aparejos de la nave.

One does not need to know Spanish to realize that the examples of the divergence in these two descriptions are numerous.

Some homonyms in a given language are translated into another language by non-homonymous lexemes, like the Spanish *antigüedad*.

In other cases, a set of homonyms in a given language is translated into a similar set of homonyms in the other language, like the Spanish *plato* when translated into the English *dish* (two possible interpretations are ‘portion of food’ and ‘kind of crockery’).

Thus, bilingual considerations sometimes help to find homonyms and distinguishing their meanings, though the main considerations should be deserved to the inner facts of the given language.

It can be concluded that synonymy and homonymy are important and unavoidable properties of any natural language. They bring

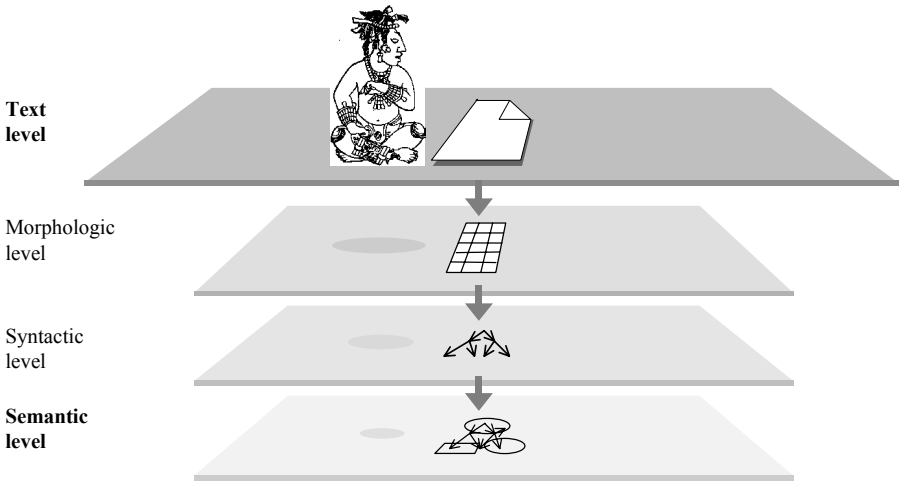

FIGURE IV.10. *Levels of linguistic representation.*

many heavy problems into computational linguistics, especially homonymy.

Classical lexicography can help to define these problems, but their resolution during the analysis is on computational linguistics.

MULTISTAGE CHARACTER OF THE MEANING ⇔ TEXT TRANSFORMER

The ambiguity of the Meaning ⇔ Text mapping in both directions, as well as the complicated structure of entities on both ends of the Meaning ⇔ Text transformer make it impossible to study this transformer without dividing the process of transformation into several sequential stages.

Existence of such stages in natural language is acknowledged by many linguists. In this way, intermediate levels of representation of the information under processing are introduced (see Figure IV.10), as well as partial transformers for transition from a level to an adjacent (see Figure IV.11).

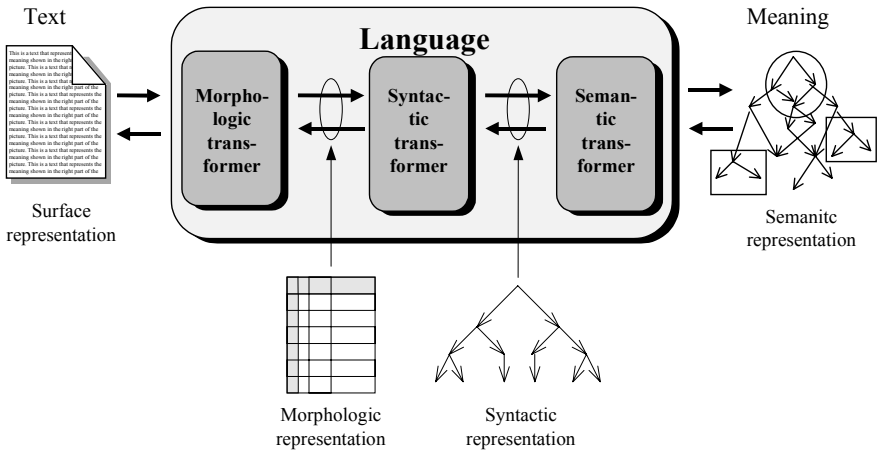

FIGURE IV.11. *Stages of transformation.*

Two intermediate levels are commonly accepted by all linguists, with small differences in the definitions, namely the morphologic and syntactic ones.

In fact, classical general linguistics laid the basis for such a division before any modern research. We will study these levels later in detail.

Other intermediate levels are often introduced by some linguists so that the partial transformers themselves are divided into subtransformers. For example, a surface and a deep syntactic level are introduced for the syntactic stage, and deep and a surface morphologic level for the morphologic one.

Thus, we can imagine language as a multistage, or multilevel, Meaning ⇔ Text transformer (see Figure IV.12).

The knowledge necessary for each level of transformation is represented in computer dictionaries and computer grammars (see Figure IV.13). A *computer dictionary* is a collection of information on each word, and thus it is the main knowledge base of a text processing system. A *computer grammar* is a set of rules based on common properties of large groups of words. Hence, the grammar rules are equally applicable to many words.

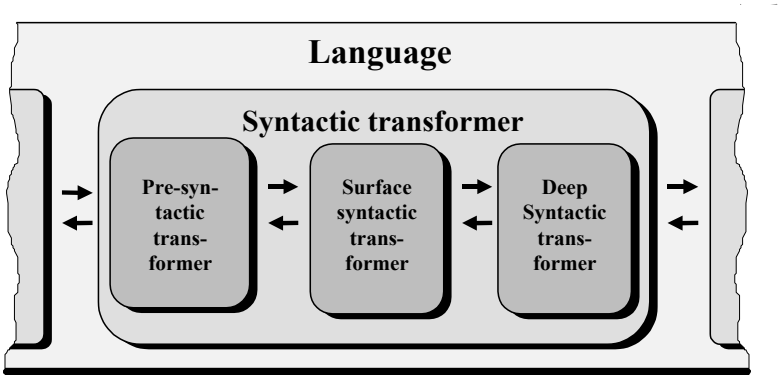

FIGURE IV.12. *Interlevel processing.*

Since the information stored in the dictionaries for each lexeme is specified for each linguistic level separately, program developers often distinguish a morphologic dictionary that specifies the morphologic information for each word, a syntactic dictionary, and a semantic dictionary, as in Figure IV.13.

In contrast, all information can be represented in one dictionary, giving for each lexeme all the necessary data. In this case, the dictionary entry for each lexeme has several *zones* that give the properties of this lexeme at the given linguistic level, i.e., a morphologic zone, syntactic zone, and semantic zone.

Clearly, these two representations of the dictionary are logically equivalent.

According to Figure IV.13, the information about lexemes is distributed among several linguistic levels. In Text, there are only wordforms. In analysis, lexemes as units under processing are involved at morphologic level. Then they are used at surface and deep syntactical levels and at last disappeared at semantic level, giving up their places to semantic elements. The latter elements conserve the meaning of lexemes, but are devoid of their purely grammatical properties, such as part of speech or gender. Hence, we can conclude that there is no level in the Text $\Rightarrow$ Meaning transformer, which could be called lexical.

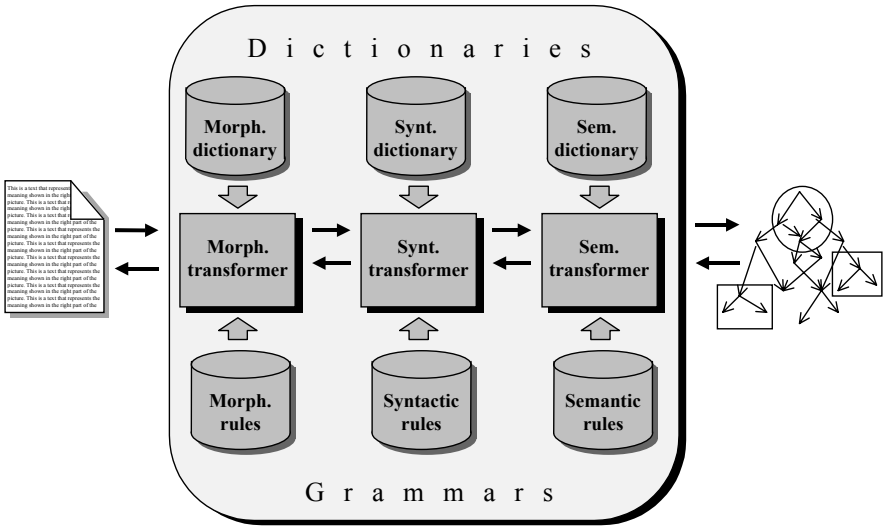

FIGURE IV.13. *The role of dictionaries and grammars in linguistic transformations.*

TRANSLATION AS A MULTISTAGE TRANSFORMATION

The task of translation from one natural language to another is a good illustration of multistage transformation of linguistic information.

Suppose there is a text in a language *A* that is to be translated into language *B*. As we have already argued, word-by-word translation leads to very poor and useless results. To translate the text with highest possible quality, the following stages of transformation are necessary:

- *First stage of analysis* starts from the source text in the language *A* and gives its morphologic representation specific for language *A*.

- *Second stage of analysis* starts from the morphologic representation and gives the syntactic representation specific for language *A*.
- *Third stage of analysis* starts from the syntactic representation and gives some level of semantic representation. The latter can be somewhat specific to language *A*, i.e., not universal, so that additional intra-level operations of "universalization" of semantic representation may be necessary.

The problem is that currently it is still not possible to reach the true semantic representation, i.e., the true level of Meaning, consisting of the universal and thus standard set of semes. Therefore, all practical systems have to stop this series of transformations at some level, as deep as possible, but not yet at that of universal Meaning.

- *The transfer stage* replaces the labels, i.e., of the conventional names of the concepts in language *A,* to the corresponding labels of language *B*. The result is the corresponding quasi-semantic level of representation in language *B*. In some cases, additional, more complex intra-level operations of "localization" are necessary at this stage.
- *First stage of synthesis* starts from the quasi-semantic representation with few features specific for the language *B*, and gives the syntactic representation quite specific for this language.
- *Second stage of synthesis* starts from the syntactic representation, and gives the morphologic representation specific for language *B*.
- *Third stage of synthesis* starts from the morphologic representation, and gives the target text in language *B*.

In the initial stages, the transformations go to the deep levels of the language, and then, in the last stages, return to the surface, with the ultimate result in textual form once more. The deeper the level reached, the smaller the difference between the representations of this level in both languages *A* and *B*. At the level of Meaning, there is no difference at all (except maybe for the labels at semes). The

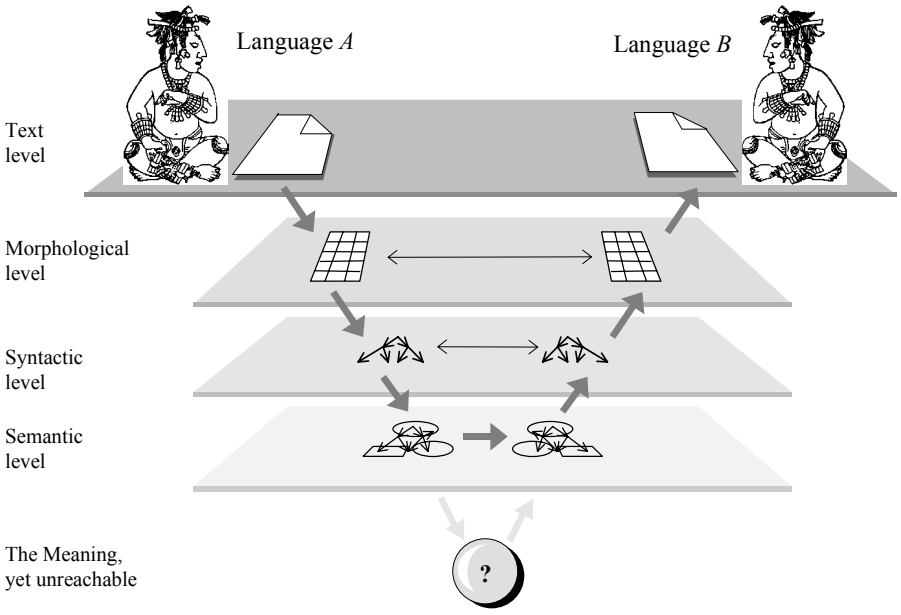

FIGURE IV.14. *Translation as multistage transformation.*

deeper the level reached during the transformations, the smaller the differences that have to be ignored, and the better the quality of translation (see Figure IV.14). This scheme shows only the general idea of all these representations.

The given scheme works for an arbitrary pair of natural languages. However, if the two languages are very similar in their structure, the deeper stages of the transformation might not be necessary.

For example, if we translate from Spanish into Portuguese, then, because these two languages differ mainly in their lexicon, it can be sufficient to use only the first stage of analysis and the last stage of synthesis, just replacing each Spanish word by the corresponding Portuguese one on the morphologic level.

In Figure IV.14 this would correspond then to the "horizontal" transition directly on this level.

TWO SIDES OF A SIGN

The notion of *sign,* so important for linguistics, was first proposed in a science called *semiotics*. The sign was defined as an entity consisting of two components, the *signifier* and the *signified*. Let us first consider some examples of *non-linguistic* signs taken from everyday life.

- If you see a picture with a stylized image of a man in a wheelchair on the wall in the subway, you know that the place under the image is intended for disabled people. This is a typical case of a sign: the picture itself, i.e., a specific figure in contrasting colors, is the signifier, while the suggestion to assist the handicapped persons is the signified.
- Twenty Mexican pesos have two equally possible signifiers: a coin with a portrait of Octavio Paz and a piece of paper with a portrait of Benito Juárez. The signified of both of them is the value of twenty pesos. Clearly, neither of them *has* this value, but instead they *denote* it. Thus, they are two different but synonymous signs.
- Raising one's hand (this gesture is the signifier) can have several different signifieds, for instance: (1) an intention to answer the teacher's question in a classroom, (2) a vote for or against a proposal at a meeting, (3) an intention to call a taxi in the street, etc. These are three different, though homonymous, signs.

LINGUISTIC SIGN

The notion of *linguistic sign* was introduced by Ferdinand de Saussure. By linguistic signs, we mean the entities used in natural languages, such as morphs, lexemes, and phrases.

Linguistic signs have several specific properties, the most obvious of which is that they are to be combined together into larger

signs and each one can in turn consist of several smaller signs. Natural language can be viewed as a system of linguistic signs.

As another property of linguistic sign, its signifier at the surface level consists of elementary parts, phonetic symbols in the phonetic transcription or letters in the written form of language. These parts do not have any signified of their own: a letter has no meaning, but certain strings of letters do have it.

We have already mentioned other notation systems to represent words of natural language, such as hieroglyphic writing. Each hieroglyph usually has its own meaning, so a hieroglyph *is* a linguistic sign. The gestures in the sign language for deaf people in some cases do have there own meanings, like hieroglyphs, and in other cases do not have any meaning of their own and serve like letters.

LINGUISTIC SIGN IN THE MMT

In addition to the two well-known components of a sign, in the Meaning ⇔ Text Theory yet another, a third component of a sign, is considered essential: a record about its ability or inability to combine with other specific signs. This additional component is called *syntactics* of the linguistic sign. For example, the ending morph *-ar* for Spanish infinitives has the same signified as *-er* and *-ir*. However, only one of them can be used in the wordform *hablar*, due to the syntactics of these three endings, as well as to the syntactics of the stem *habl-*. We can imagine a linguistic sign as it is shown on Figure IV.15.

Thus, syntactics of linguistic signs helps to choose one correct sign from a set of synonymous signs, to be used in a specific context. For example, we choose the ending *-ar* for the stem *habl-* since it does accept just it. Beyond that, syntactics of linguistic signs helps us to disambiguate the signs, i.e., to decide which of the homonymous signs is used in a specific context.

For the extralinguistic example given above, the classroom is the context, in which we interpret the raising of one's hand as the intention to say something rather than to call a taxi.

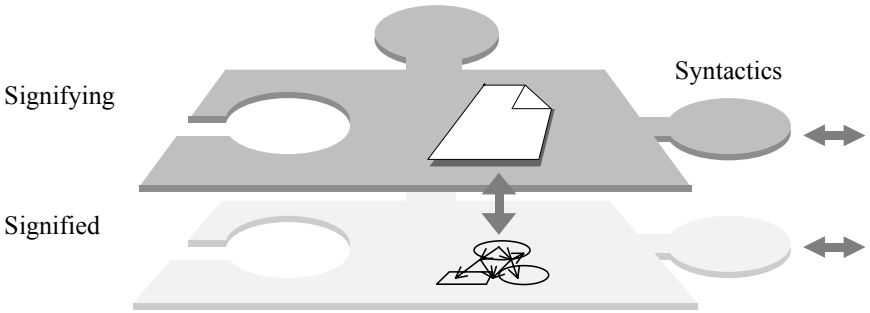

FIGURE IV.15. *Syntactics of a linguistic sign.*

Similarly, the presence of the Spanish stem *salt-* is sufficient to interpret the *-as* ending as present tense, second person, singular in *saltas*, rather than feminine, plural as in the presence of the stem *roj-*: *rojas*.

LINGUISTIC SIGN IN HPSG

In Head-driven Phrase Structure Grammar a linguistic sign, as usually, consists of two main components, a signifier and a signified. The signifier is defined as a phoneme string (or a sequence of such strings). Taking into account the correspondence between acoustic and written forms of language, such signifier can be identified as conceptually coinciding with elements of Text in the MTT.

As to the signified, an object of special type SYNSEM is introduced for it. An object of this type is a structure (namely, a labeled tree called *feature structure*) with arcs representing features of various linguistic levels: morphologic, syntactic, and semantic, in a mixture. For a minimal element of the text, i.e., for a wordform, these features show:

- How to combine this wordform with the other wordforms in the context when forming syntactic structures?
- What logical predicate this word can be a part of?
- What role this word can play in this predicate?

Simple considerations show that SYNSEM in HPSG unites the properties of Meaning and syntactics of a sign as defined in the framework of the MTT, i.e., SYNSEM covers syntactics plus semantics. Hence, if all relevant linguistic facts are taken into consideration equally and properly by the two approaches, both definitions of the linguistic sign, in HPSG and MTT, should lead to the same results.

ARE SIGNIFIERS GIVEN BY NATURE OR BY CONVENTION?

The notion of sign appeared rather recently. However, the notions equivalent to the signifier and the signified were discussed in science from the times of the ancient Greeks. For several centuries, it has been debated whether the signifiers of things are given to us by nature or by human convention.

The proponents of the first point of view argued that some words in any language directly arose from *onomatopoeia*, or sound imitation. Indeed, we denote the sounds produced by a cat with verb *mew* or *meow* in English, *maullar* in Spanish, and *miaukat'* in Russian. Hence, according to them, common signifiers lead to similar signifieds, thus creating a material link between the two aspects of the sign.

The opponents of this view objected that only a minority of words in any language takes their origin from such imitation, all the other words being quite arbitrary. For example, there is no resemblance between Spanish *pared*, English *wall*, and Russian *stena*, though all of them signify the same entity. Meanwhile, the phonetic similarity of the German *Wand* to the English *wall*, or the French *paroi* to the Spanish *pared*, or the Bulgarian *stena* to the Russian *stena* are caused by purely historic reasons, i.e., by the common origin of those pairs of languages.

The latter point of view has become prevalent, and now nobody looks for material links between the two sides of linguistic signs. Thus, a linguistic sign is considered a human convention that assigns specific meanings to some material things such as strings of

letters, sequences of sounds, pictures in hieroglyphic writing or gestures in sign language.

GENERATIVE, MTT, AND CONSTRAINT IDEAS IN COMPARISON

In this book, three major approaches to linguistic description have been discussed till now, with different degree of detail: (1) generative approach developed by N. Chomsky, (2) the Meaning ⇔ Text approach developed by I. Mel'čuk, and (3) constraint-based approach exemplified by the HPSG theory. In the ideal case, they produce equivalent results on identical language inputs. However, they have deep differences in the underlying ideas. In addition, they use similar terminology, but with different meaning, which may be misleading. In this section, we will compare their underlying ideas and the terminology. To make so different paradigms comparable, we will take only a bird's-eye view of them, emphasizing the crucial commonalities and differences, but in no way pretending to a more deep description of any of these approaches just now.

Perhaps the most important commonality of the three approaches is that they can be viewed in terms of linguistic signs. All of them describe the structure of the signs of the given language. All of them are used in computational practice to find the Meaning corresponding to a given Text and vice versa. However, the way they describe the signs of language, and as a consequence the way those descriptions are used to build computer programs, is different. *Generative idea*. The initial motivation for the generative idea was the fact that describing the language is much more difficult, labor-consuming, and error-prone task than writing a program that uses such a description for text analysis. Thus, the formalism for description of language should be oriented to the process of describing and not to the process of practical application. Once created, such a description can be applied somehow.

Now, what is to describe a given language? In the view of generative tradition, it means, roughly speaking, to list all signs in it (in fact, this is frequently referred to as generative idea). Clearly, for a

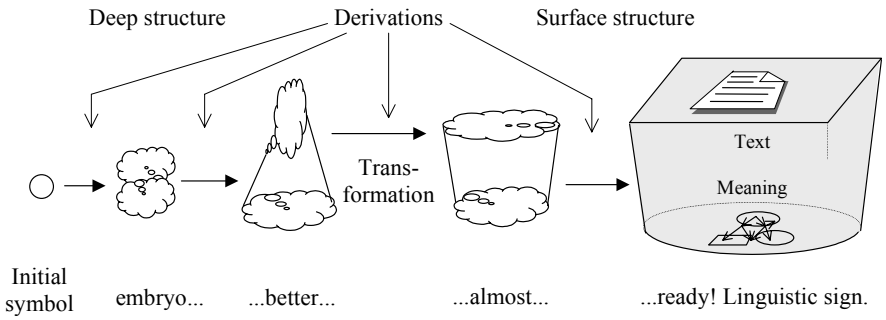

FIGURE IV.16. *Generative idea.*

natural language it is impossible to literally list all signs in it, since their number is infinite. Thus, more strictly speaking, a generative grammar describes an algorithm that lists only the correct signs of the given language, and lists them all—in the sense that any given sign would appear in its output after a some time, perhaps long enough. The very name *generative* grammar is due to that it describes the process of *generating* all language signs, one by one at a time.

There can be many ways to generate language signs. The specific kind of generative grammars suggested by N. Chomsky constructs each sign gradually, through a series of intermediate, half-finished sign "embryos" of different degree of maturity (see Figure IV.16). All of them are built starting from the same "egg-cell" called *initial symbol*, which is not a sign of the given language. A very simple example of the rules for such gradual building is given on the pages 35 to 39; in this example, the tree structure can be roughly considered the Meaning of the corresponding string.

Where the infinity of generated signs comes from? At each step, called *derivation*, the generation can be continued in different ways, with any number of derivation steps. Thus, there exist an infinite number of signs with very long derivation paths, though for each specific sign its derivation process is finite.

However, all this generation process is only imaginable, and serves for the formalism of description of language. It is not—and is

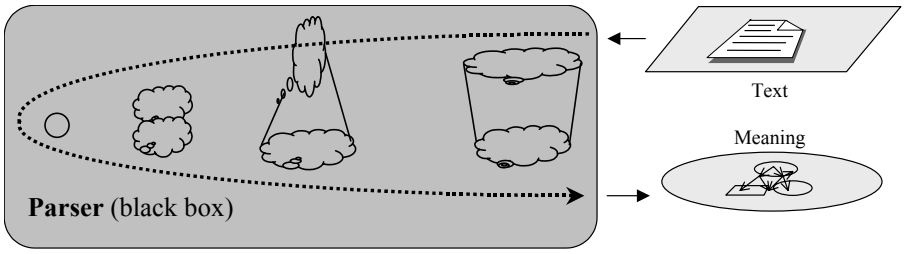

FIGURE IV.17. *Practical application of the generative idea.*

not intended to be—applied in practice for the generation of an infinitely long list of language expressions, which would be senseless. The use of the description—once created—for passing from Text to Meaning and vice versa is indirect. A program called *parser* is developed by a mathematician (not a linguist) by means of automatic "reversing" of the original description of the generative process.

This program can answer the questions: *What signs would it generate that have the given Text as the signifier? What signs would it generate that have the given Meaning as signified?* (See Figure IV.17.)

The parser does not really try to generate any signs, but instead solves such an equation using the data structures and algorithms quite different from the original description of the generating process.

The result produced by such a black box is, however, exactly the same: given a Text, the parser finds such Meaning that the corresponding sign belongs to the given language, i.e., *would* be generated by the imaginable generation algorithm. However, the description of the imaginable generation process is much clearer than the description of the internal structures automatically built by the parser for the practical applications.

Meaning ⇔ Text idea. As any other grammar, it is aimed at the practical application in language analysis and synthesis. Unlike generative grammar, it does not concentrate on enumeration of all possible language signs, but instead on the laws of the correspondence between the Text and the Meaning in any sign of the given

language. Whereas for a given text, a generative grammar can answer the question *Do any signs with such Text exist, and if so, what are their Meanings?*, a the MTT grammar only guarantees the answer to the question *If signs with such Text existed, what would be their Meanings?*

In practice, the MTT models usually *can* distinguish existing signs from ungrammatical ones, but mainly as a side effect. This makes the MTT models more robust in parsing.

Another idea underlying the MTT approach is that linguists are good enough at the intuitive understanding of the correspondence between Texts and Meanings, and can describe such correspondences directly. This allows avoiding the complications of generative grammars concerning the reversion of rules. Instead, the rules are applied to the corresponding data structures directly as written down by the linguist (such property of a grammar is sometimes called *type transparency* [47]). Direct application of the rules greatly simplifies debugging of the grammar. In addition, the direct description of the correspondence between Text and Meaning is supposed to better suite the linguistic reality and thus results in less number of rules.

Similarly to the situation with generative grammars, there can be many ways to describe the correspondence between Text and Meaning. The specific kind of the MTT grammars suggested by I. Mel'čuk describes such a correspondence gradually, through many intermediate, half-finished almost-Meanings, half-Meanings, half-Texts, and almost-Texts, as if they were located inside the same sign between its Meaning and Text (see Figure IV.18).

Since the MTT and the generative approach developed rather independently, by accident, they use similar terms in quite different and independent meanings. Below we explain the differences in the use of some terms, though these informal explanations are *not* strict definitions.

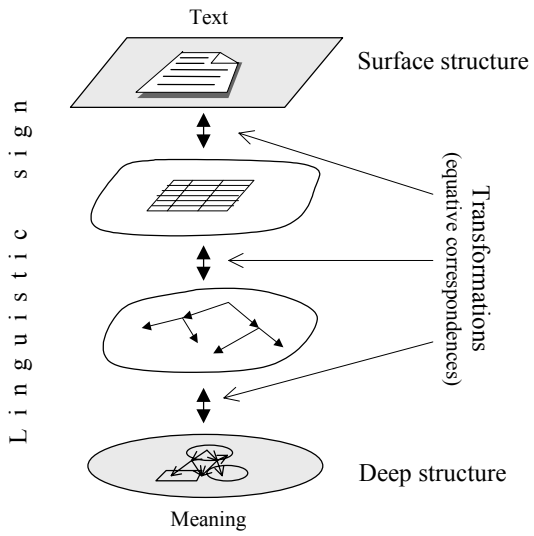

FIGURE IV.18. *Meaning ⇔ Text idea.*

- In generative grammar (see Figure IV.16):
  - *Transformation*: a term used in early works by N. Chomsky for a specific kind of non-context-free derivation.
  - *Deep structure*, in the transformational grammar, is a half-finished sign with a special structure to which a transformation is applied to obtain a "readier" sign. It is nearer to the initial symbol than the surface structure.
  - *Surface structure* is a half-finished sign obtained as the result of the transformation. It is nearer to the ready sign than the deep structure.
  - *Generation* is used roughly as a synonym of derivation, to refer to the process of enumeration of the signs in the given language.
- In the MTT (see Figure IV.18):
  - *Transformation* is sometimes used for equative correspondences between representations on different levels.

- *Deep structure* concerns the representation nearer to Meaning.
- *Surface structure* concerns the representation nearer to Text.
- *Generation* (of text) is used sometimes as a synonym of synthesis, i.e., construction of Text for the given Meaning.

Constraint-based idea. Similarly to the generative grammar, a constraint-based grammar describes what signs exist in the given language, however not by means of explicit listing (generation) of all such signs, but rather by stating the conditions (constraints) each sign of the given language must satisfy.

It can be viewed as if it specified what signs do *not* exist in the given language: if you remove one rule (generation option) from a generative grammar, it will generate *less* signs. If you remove one rule (constraint) from a constraint-based grammar, it will allow *more* signs (i.e., allow some signs that really are ungrammatical in the given language). Hence is the name *constraint-based*. (See also page 44.)

Since constraint-based grammars do not use the generation process shown on Figure IV.16, their rules are applied within the same sign rather than to obtain one sign from another (half-finished) one.

This makes it similar to the MTT. Indeed, though the constraint-based approach was originated in the generative tradition, modern constraint-based grammars such as HPSG show less and less similarities with Chomskian tradition and more and more similarity—not in the formalism but in meaningful linguistic structures—with the MTT.

A constraint-based grammar is like a system of equations. Let us consider a simple mathematical analogy.

Each sheet of this book is numbered at both sides. Consider the side with even numbers. Looking at the page number, say, 32, you can guess that it is printed on the 16-th sheet of the book. Let what you see be Text and what you guess be Meaning; then this page number corresponds to a "sign" <32, 16>, where we denote <T, M> a sign with the Text T and Meaning M. In order to describe such a

"language", the three approaches would use different mathematical constructions (of course, in a very rough analogy):

- Generative grammar is like a *recurrent formula*: The sign <2, 1> (analogue of the initial symbol) belongs to this "language", and if <x, y> belongs to it, then <x + 2, y + 1> belongs to it (analogue of a generation rule). Note that some effort is needed to figure out from this description how to find a sheet number by a page number.

- The MTT grammar is like an *algorithm*: given the page number x, its sheet number is calculated as $x/2$; given a sheet number y, its page number is calculated as $2 \times y$. Note that we have made no attempt to describe dealing with, or excluding of, odd page numbers x, which in fact do not belong to our "language."

- Constraint-based grammar is like an *equation* or *system of equations*. Just those signs belong to our "language," for which $x = 2y$. Note that this description is the most elegant and simple, completely and accurately describes our "language," and requires less reversing effort for practical application than the first one. However, it is more complex than the second one.

Constraint-based idea is a very promising approach adopted by the majority of contemporaneous grammar formalisms. Probably with time, the linguistic findings of the MTT will be re-formulated in the form of constraint-based rules, possibly by a kind of merging of linguistic heritage of the MTT and formalisms developed in frame of HPSG. However, for the time being we consider the MTT more mature and thus richer in detailed description of a vast variety of linguistic phenomena. In addition, this approach is most directly applicable, i.e., it does not need any reversing.

As to the practical implementation of HPSG parsers, it is still an ongoing effort at present.

CONCLUSIONS

The definition of language has been suggested as a transformer between the two equivalent representations of information, the Text, i.e., the surface textual representation, and the Meaning, i.e., the deep semantic representation. This transformation is ambiguous in both directions: a homonymous Text corresponds to several different Meanings, and several synonymous Texts correspond to the same Meaning.

The description of the transformation process is greatly simplified by introducing intermediate levels of information representation, of which the main are morphologic and syntactic. At each level, some of the problems arising from synonymy and homonymy can be solved.

The general definitions of linguistic sign in Meaning ⇔ Text Theory and in Head-driven Phrase Structure Grammar turned out to be in essence equivalent.