COMPUTATIONAL LINGUISTICS

VI

COMPUTING CENTRE OF THE HUNGARIAN ACADEMY OF SCIENCES
BUDAPEST, 1967

COMPUTATIONAL LINGUISTICS
VI.

COMPUTING CENTRE OF THE HUNGARIAN ACADEMY OF SCIENCES
BUDAPEST, 1967

Felelős kiadó: FREY TAMÁS

682177 MTA KESZ Sokszorosító

C O N T E N T S

R E V I E W

SEMANTIC LINKS OF VERBAL ENTRIES[1]

L. Elekfi

1. The Exact Investigation of Semantic Interrelations

1.1. The dictionary explanation of a content word is correct if it can be substituted for the entry word. If substitutable, it is e q u i v a l e n t to the entry item. Equivalence does not mean, of course, that the explanation is in every respect equivalent to the entry. In most instances it is not identical with it as regards stylistic value and/or frequency. However, its denotational value, its objective conceptual meaning is identical with that of the entry word. In this sense the text of the explanation is a s y n o n y m of the entry. More particularly, it may be a simple synonym /proper/, that is, a synonymous word, or it may be a paraphrase of which the most exact version is a d e f i n i t i o n[2].

Explanation or definition, as an operation, is a kind of t r a n s f o r m a t i o n. Not a grammatical transformation as expounded by Harris and Chomsky, in the course of which the words or word stems, i.e. lexical morphemes, remain intact and only the grammatical construction, some grammatical morphemes are altered, but a semantic transformation in which a word is replaced by another word or word group in a way that the semantic content of the replaced word should remain the same or that any possible resultant shift should be kept to a minimum. This kind of transformation may be called synonymic transformation in comparision to one in which the meaning of the linguistic unit is changed /e. g. negation, substitution by pronouns/; the latter may be termed, if a name is needed, altering transformation.

The logical type of definition consists of two elements: one is the genus proximum /this is identical with the defined word in word-class membership and denotes a superordinate concept; broadly it is a synonym of the former/, the other is the differentia specifica /this is of a word-class different from that of the entry: an adjective alongside a noun, an adverb alongside an adjective, etc./ and constitutes the restrictive part of the definition. Both the complete definition and the genus proximum stand in some synonymic relationship with the word explained or defined, but whereas this relationship of the full definition is characterized by /relative/ equivalence, that is, mutual substitutability, that of the genus proximum is characterized by subsumption /and superordination resp./, that is, by a one-way replaceability, in other words, by logical implication.[3]

1.2. The Hungarian Explanatory Dictionary usually interprets a word with the help of a single synonym if the word in question possesses a particular stylistic value, or if it is not current in standard language but is confined to some region, to a social or occupational group or to a certain context, further if there is in the standard language a word equivalent to it as far as its objective conceptual denotation is concerned. In such cases the way from the entry to the explanatory word is of a socio-linguistic and stylistic nature: it coincides with the way from the less common word of more limited use to the standard word of general currency. In the case of explanation by definition or paraphrase the way from the entry word to the main item of the explanation, to the genus proximum is more of a conceptual, logical nature: it leads from a more specialized concept to a more general one.

If it is desirable to group items belonging to the same word-class according to concepts, it is expedient to adopt a method already applied in the dictionary explanations. The word used as genus proximum in the explanation generally denotes a concept of wider range than the entry word itself. In this way we get from the species to the genus, which often figures as the leading word in the explanation of several entries, but when we look at the explanation of the word denoting the genus, we find a leading word denoting a concept of a still wider range, a still higher class, and so on and so forth up to a certain limit. E. g. the explanation /somewhat shortened/ of meaning 1 of the noun legény /lad, young man/ is as follows: "Felserdült, házasságra érett nőtlen férfi" /grown-up, single man of marriageable age/. The genus proximum of this definition is the noun qualified by the adjectives: férfi /man/. The explanation of férfi /man/ is "Himnemű felnőtt személy" /male adult person/. The genus proximum of személy /person/ is lény /being/. The leading word of the explanation of lény /being/ is egyed /individual entity/[4], that of egyed is dolog /thing/.

By going through the explanations of the nouns in the dictionary in this way we obtain e x p l a n a t o r y s t r i n g s consisting of 4-5, occasionally more, items, with a word with the narrowest and most specialized meaning at one end and a word of the most general meaning at the other. The more general the meaning of a noun is the more frequently will it occur as the leading noun in the explanation of nouns.[5]

On the basis of the foregoing the explanatory string of meaning 1 of legény /lad, young man/ will be:
legény -- férfi -- személy -- lény -- egyed -- dolog. -
From the explanation of dolog, however, the mechanical tracing of the explanatory leading noun does not yield a concept of higher order or greater generality. Under the comprehensive explanation of the semantic ramifications of this polysemantic word we find the following synonyms:

I. "munka" /work/, II. "cselekedet" /doing, action/, III. "körülmény" /circumstance/, IV. "tárgy" /object, thing/. Two things stand out clear from this: 1/ the leading word of the explanation should not always be taken in its first or primary meaning. Thus the word dolog appearing in the explanation of egyed stands closest to the range of meanings detailed under dolog IV., although it does not quite fit in there either. -- 2/ When the editors of the dictionary failed to find a word with a meaning wider than the entry item, then they had to be content with a more or less appropriate synonym, that is to say, they had to give up a proper definition. It follows from this, on the other hand, that, when going through an explanatory string we come to a break like this, then we have already arrived at the words of the greatest conceptual range and most general meaning.

1.3. Completeness would require that we should explore the explanatory strings of all the meanings of all the words to be entered, but for a start we should be satisfied with setting out from the first meaning of the individual items.

2. The Semantic Categories of Verbs

2.1. Assigning verbs to their conceptual categories is not the same as with nouns. The Explanatory Dictionary explains a verb with a synonym, usually a periphrastic one, with the help of a synonymous verb whose range of usage and meaning is roughly coterminous with that of the entry verb or is only minimally larger.

2.2. Assignment to more general concepts and categories is carried out by the Explanatory Dictionary not in the explanation of verbs but noun derivatives in -ás, -és /and some derivatives in -tal, -tel, rarely -at, -et/. Therefore it is advisable to start from the explanation of verbs but noun derivatives in -ás, -és /and some derivatives in -tal, -tel, rarely -at, -et/. Therefore it is advisable to start from the expalanation the relevant nomen actionis when looking for the main groups. E. g. járás /going/: "a jár igével kifejezett c s e l e k v é s" /action expressed by the verb 'go'/; verés /beating/: "az a c s e l e k v é s, hogy vkit, vmit vernek" /the action that sy or sg is being beaten/; húzás /drawing/, ugatás /barking/, elmenetel /going away/: "cselekvés" /action/; tördelés /breaking into pieces/, halászat /fishing/: "tevékenység" /activity/; átköltés /rendering into another poetic form/: "eljárás" /treatment, procedure/; felhősödés /clouding over/, levés /becoming/: "folyamat" /process/; hallás /hearing/: "érzékelési folyamat" /perceptual process/; csattanás /a [single] clap/: "történés" /event/; mennydörgés /thunder/, lobbanás /a [single] flash/: "jelenség" /phenomenon/; villámlás /lightning/: "fényjelenség" /phenomenon of light/; csikorgás /creaking/; "hangjelenség" /phenomenon of sound/; állás /standing/: "állapot" /state/; fájás /pain/: "érzés" /sensation/; tegeződés /"thou-ing"/: "érintkezési mód két v. több személy közt" /manner of intercourse between two or more persons/; maradás /remaining/: "az a tény, hogy vki, vmi marad" /the fact that sy or sg remains/; létel /existence/: "az a tény, hogy vki, vmi van, létezik" /the fact that sy or sg is, exists/; vonatkozás /relating/: "viszony" /relationship/. As a rule the Explanatory Dictionary gives its more detailed explanation of the nomen actionis concerned only after assigning it to some such general category. Less frequent are such names of action in explaining which the Dictionary, instead of designating the main category, uses a word denoting a relatively narrower concept /though a more general one than the deverbal noun/; e. g. adózás /taxation/: "a kivetett adó/k/ rendszeres fizetése" /the regular payment of the tax/es/ imposed/.

If the conceptual content of a verb can be assigned to more than one category, the summary explanation of the nomen actionis refers to more than one general notion of occurrence. E. g. festés /painting/: "cselekvés, művelet" /action, operation/; lépés /step/, taszítás /repulsion/: "cselekvés, mozdulat" /action, movement/; csaholás /barking/: "cselekvés, hangadás" /action, emission of sound/; ugrás /leaping/: "cselekvés, mozgás" /action, motion/; gyulladás /inflammation/: "folyamat, mozzanat" /process, momentum/; születés /birth/: "folyamat, történés" /process, occurrence/; felháborodás /indignation/, gyűlölködés /animosity/: "lelkiállapot, magatartás" /state of mind, attitude/; csodálkozás /wonderment/: "folyamat, állapot" /process, state/; viselkedés /behaviour, conduct/: "cselekvés, ill. magatartás" /action and attitude respectively/; eladás /selling/: "az elad igével kifejezett cselekvés, ill. folyamat" /action or process expressed by the verb "sell"/; könnyítés /easing/: "cselekvés, eljárás, rendelkezés" /action, procedure, provision/; hizlalás /fattening/: "tevékenység, eljárás" /activity, procedure/; gazdálkodás /farming/: "tevékenység, foglalkozás" /activity, occupation/; törés /breaking/: "cselekvés, tevékenység" /action, activity/; dicsekedés /boasting/: "cselekvés, megnyilvánulás" /action, manifestation/; dicsőítés /glorification/: "cselekvés, megnyilatkozás" /action, expression/; kutatás /research/: "cselekvés, eljárás, munka" /action, procedure, work/; átruházás /transference, relegation/: "cselekvés, /jogi/ eljárás, intézkedés" /action, /legal/ procedure, provision/; nyaralás /spending one's summer holidays swhere/: "állapot, időtöltés" /state, passing of time/; tudás /knowledge/: "állapot, képesség" /state, faculty/; lejtés /slope, sloping/: "helyzet, ill. mozgás" /position, resp. motion/.

In analyzing the explanations we must, of course, reckon also with some accidental features of the explanatory texts, namely, with the fact that many of the definitions might have been differently phrased by the editorial staff. However during the investigation of the mass of data, /of which only a few samples are given here/ there have nevertheless emerged the regularities of the explanations of the entries belonging to identical categories, which the editors followed partly consciously, partly guided by their intuitive linguistic competence.

Another fact that may not be held altogether accidental is that the Dictionary defines several names of action by more than one synonyms in coordination, since these nouns--as well as their basic verbs --really fluctuate between two or more categories of occurrence, or the comprehensive explanation takes into account several of the verb's meanings at once.

In the explanations of the nomina actionis the synonyms representing a higher order of concept /genus proximum/ are often placed in coordination only syntactically and as regards conceptual hierarchy one is subordinated to the other /and in such instances naturally the formal subordination conceals rather an explanatory apposition this is like a construction as if the conjunction namely stood before the second item/. If it is desired to make a list of the principal categories of the names of action on the basis of the Explanatory Dictionary, then it is possible to exclude these nouns denoting subordinate concepts from the start, since it is obvious that they merely designate subcategories. By this means we can subsume the categories "operation", "movement", "emission of sound", "manifestation", "expression", etc. under that of "action". Similarly the category of "activity" will subsume "occupation" and "faculty" will be a subordinate synonym[6] of "state", while "instance" of "process".

Occasionally there come together two explanatory synonyms which are in purely coordinate relationship, as in the case of ugrás /leaping/ "cselekvés, mozgás" /action, motion/, where the basic verb really denotes both action and motion, and the action denoted by the verb is at the same time motion. - In other cases two synonyms given by way of explanation stand in a relationship that both express the same idea from two different points of view. This is exemplified by születés /birth/ "folyamat, történés" /process, occurrence/. If we look at the explanation of folyamat we do not find an explicit reference to történés, only to esemény /event/; in the explanation of esemény then we find történés; on the other hand, the explanation of történés includes the word folyamat.

If in the course of semantic analysis we come across tow words denoting two such concepts which can only be interpreted in the last resort, by each other, then these words denote general conceptual categories. /In explaning words standing for concepts of lower order the Explanatory Dictionary tried to eschew such vitious circles. But even with the most general categories it deliberately avoided making two synonyms explain each other directly, and for this reason it made use of an intermediary term as much as it was possible. An instance of this is the interpolation of esemény in our last example./

Among the designations of the verbal concepts which make no use of synonyms we also find some which come under a higher category. Thus, for instance, eljárás /procedure/ comes under the concept of "activity", but the explanation of "procedure" and "activity" contain "action", which, in turn, is explained by such a complicated paraphrase that, if for no other reason, it may be looked upon as a general category.

If two explanatory words follow each other as alternatives--usually separated by ill. /or. resp./, this is a definite pointer that the verbal idea belongs to two different categories. E. g. viselkedés /behavior, action, resp. attitude, conduct/: "cselekvés, ill. magatartás".

The analysis of all the other examples might be continued along these lines, but now instead of doing so let us see the result that we would obtain in this way: which are the designations expressing the most general categories of words of the nomen actionis type?

So far we have set up the following categories: "action", "state", "process", "occurrence". Carrying on this kind of exploration we could add to this list items such as "manner", "phenomenon", "motion", "sensation", "fact", "relationship".

2.3. Let us now turn our attention to the explanations of the verbs themselves to be found in the Dictionary. How do the meaning equivalents of the individual verbs reflect which caterory they belong to? Let us first see those of the verbs underlying the nomina actionis examined above which can be uniquely assigned to some one of the principal categories. The verbs of several meanings will here be considered only on the basis of the explanation of their primary meaning or the first main group of their meanings. The explanations will be occasionall y abbreviated. /Departures from this practice will be indicated by figures appearing after the entry word./

A c t i o n: jár /goes/ "sorozatosan lép és így változtatja helyét" /steps successively and changes his[7] place in this way/ | ver /beats/ "gyors egymásutánban többször v. hosszabb ideig üt" /hits in rapid succession several times or for a longer period/ | huz /draws , pulls/ "maga felé vagy saját mozgása irányában mozgat" /moves towards himself or in the direction of his own movement/ | ugat /barks/ " ‹kutya, farkas, róka, ritk. más állat› szaggatottan több egymást követő éles hangot ad" /‹dog, wolf, fox, less commonly some other animal› gives several sharp noises in succession with interruptions/ | elmegy /goes away/ "eltávozik, eljut" /departs, gets swhere/. -- /Activity:/ tördel /breaks into pieces/ "vmit egymás után több kisebb darabra tör" /breaks sg into several smaller pieces in succession/ | halászik /fishes/ "halat igyekszik fogni" /tries to catch fish/ |

gondolkozik /thinks/ "tudatos értelmi tevékenységet fejt ki" /exerts a conscious mental effort/. -- /Procedure:/ átkölt /renders into [another] poetic form/ "<költeményt> más költői formába önt" /puts <poem> into another poetic form/.

O c c u r r e n c e : csattan /claps [once]/ "<szilárd v. rugalmas tárgy vmivel összeütközve> rövid, éles hangot idéz elő" /<solid or elastic body coming into collision with sg> gives rise to a short, sharp noise/.

P h e n o m e n o n : lobban /flashes [once]/ "<meggyujtott, gyulékony anyag> hirtelen lángot fog, láng csap ki belőle" /<burning or inflammable substance> suddenly catches fire a flame darts from it/ | mennydörög /thunders/ "a villámlást követően hatalmas erejű, messze morajló hang hallatszik" /following flash of lightning a tremendous noise rumbling far is [can be] heard/ | villámlik /lightens/ "villám keletkezik" /lightning arises/ | csikorog /creaks/ "<forgó, surlódó kemény tárgy> rendsz. huzamosabb ideig, bántó, éles hangot ad" /<rotating, grating hard object> gives forth offensive, sharp noise, usually for a longer time/.

M o t i o n : folyik /flows/ "<folyadék> vmely felület mentén /lefelé/ halad" /<liquid> passes [downwards] along some surface/.

S t a t e : áll II /stands/ "nem mozog" /does not move/.

S e n s a t i o n : fáj /hurts, aches/ "vmely testrészben sajgó, gyötrő érzés jelentkezik" /in some part of the body a painful, smarting sensation appears/.

F a c t : marad /remains/ "továbbra is ott lesz, egy helyben kezd időzni" /will continue to be there, begins to stay at one place/ | van /is/ "a létezők közé tartozik" /belongs to these that exist/.

R e l a t i o n s h i p : vonatkozik /relates to sg/ "kapcsolatban van, összefügg vkivel, vmivel" /is in relationship with, is in coherence with sy, sg/.

M a n n e r : tegeződik /address each other by 'thou'/

"úgy tegez vkit, hogy az őt visszategezi" /uses the "thou"--form with sy reciprocally/.

2.4. Our aim next is to define in a more exact fashion, inductively on the basis of the texts or wording of the explanation of our specimen verbs those categories which we have already arrived at in a roundabout way. The way we have to choose is not quite a simple one because the categories we have all of us formed in our minds--categories expressed by certain words in our language--are determined by heterogeneous criteria. We cannot reduce the explanations of verbs as mechanically to elements of genus proximum and differentia specifica as we have been able to do with nouns. We have to reckon with lexical and syntactic restrictions, which may play just as great a part in the designation of the principal categories as in the explanation of any single verb, that is, in its semantic-synonymic correspondences. Thus, for instance, verbs denoting action are characterized by the circumstance that their subjects are persons, /as with our examples jár, ver, húz, elmegy/, less frequently animals /as with the verb ugat/. This observation holds even more regarding categories of activity and procedure subsumed under the concept of action /tördel, halászik, gondolkozik; átkölt/. -- Bearing all this in mind, however, we are ready now to attempt a tentative mechanical analysis of the explanations of verbs in a way that we take the verb or verbs of the explanatory texts of the entry. Where this mechanical method of tracing leads us obviously astray we shall make the appropriate corrections and adjustments.

In the explanations of verbs we only rarely meet with an explanatory verb expressing the most general verbal category, but proceeding step by step we arrive, even in this way, at the verbs of the most general meaning. Thus setting out from meaning I 1 of jár /"sorozatosan lép"/ we reach the verb lép /steps/, in the explanation of lép /"lábát ... kissé megemelve odább teszi" ; lifting a foot a little he puts it further on/ we find the verb tesz /puts, sets/, then the explanation of the appropriate meaning group II of tesz contains helyez /places, sets/, this, in turn, is explained as "vmely helyre visz és ott hagy" /takes swhere and leaves there/, of the verb visz /takes/ the explanation of its meaning group I runs "úgy tart vmit, hogy..." /holds sg in a way that.../, and proceeding along this line we get from tart /holds/ to the successive verbs kényszerít /forces/ → elér /achieves; but in meaning 1 reaches/ → /megérint /touches lightly/ → hozzányúl /touches/ → megfog /gets hold of/ → fog /holds/ → szorít /grasps/ → nyom /presses/ → hat /effects/ → működik /operates/ → dolgozik /works/ → végez /performs/ → csinál /makes/, but here we have come full circle and would start all over again as in the explanation of csinál the verbs végez, dolgozik, tesz reappear side by side with the verbs intéz /manages/ and cselekszik /acts/, new to the list, whose explanation in turn contains the verbs végez. –– Setting out from the explanatory key verb of the entry ver /beats/ we get the following series: üt /hits/ → hozzáér /comes into contact with/ → nyúl /reaches out/ → megérint /touches lightly/, from where following the previous track we arrive again at the range of meanings covered by "csinál, végez, dolgozik, tesz, cselekszik". → The explanation of húz /draws, pulls/ would result in the following string of verbs: mozgat /moves, vt/ → odahat /exerts force to the effect that.../ → fordít /turns/, etc. But it is obvious that odahat is not the synonym of mozgat /"odahat, hogy vmi mozogjon" : exerts force to the effect that sg should move, makes sg move/, but is merely the synonym of the causative suffix /-at/. When explaining a derivative verb it may happen that the explanation interprets the meaning of the suffix rather than that of the verb in question. Of course, it is possible to regard the synonym odahat of the causative suffix as a marker of category, but more closely húz belongs rather to the category of mozgat. -- Of the explanatory verbs of elmegy /goes away/ eltávozik /departs/ refers back to elmegy, while following in the track of eljut /gets swhere/ we get the sequence: megjelenik /appears/ → feltünik /shows, vi/ → előtünik /becomes visible/ → válik /becomes/ → alakul /turn into sg/ → ölt /assumes [a form]/, here, however, we are on a false scent because the latter verb expounds the meaning of the suffix -ul only, → resp. following the other synonym of válik : lesz /turns into sg/ → kezd /begins/ → megtesz /does / → véghezvisz /carries out/ → megvalósít /carries into effect/ → tesz I /does, makes/ → végez → csinál /makes/. Thus elmegy ultimately belongs to the same category as jár: to that of the verbs denoting action. → The explanation of ugat /barks/ leads to the following verbs: ad V 1b /gives [forth]/ → okoz /causes/ → előidéz /gives rise to/ → létrehoz /brings into existence/ → megvalósít /realizes/ → tesz /does, makes/. Therefore this verb, too, belongs really here.

The explanatory verb of tördel /breaks into pieces/ assigned to the sub-category of activity takes the following road: tör /breaks/ → választ /II separates/ → szétválaszt /divides/ → elválaszt /parts/ → elkülönít /segregates/ → különválaszt /isolates/ → szétszed /takes to pieces/, from here the circle of synonyms begins anew. If we take the other synonym of elkülönit: kiemel /sets apart/ → eltávolít /detaches/ → elvesz /takes away/ → tesz /II puts/. → The explanatory string of halászik /fishes/ goes like this: igyekszik /tries/ → megtesz /does/ → ... → tesz /does, makes/. → The verb átkölt /renders into [another] poetic form/ labelled as a procedure verb produces the following chain: önt /casts/ → kifejez /expresses/ → előad /sets forth/ → bemutat /presents/ → megismertet /acquaints/ → ad /gives/ → ... tesz /does, makes/.

Of the verbs denoting some process hall /hears/ leads to these verbs: érzékel /perceives/ → felfog /apprehends/ → észrevesz /notices/ → meglát /sees/ → megpillant /catches sight of/. Beginning with észrevesz we enter into a vitious circle of explanation. which, in its entirety together with the verb kezd /begins/ included in the sequence, led to a false track because what it explained was the perfective aspect of the verbal prefix of felfog /i.e. fel-/. We have to return therefore to the verb érzékel which cannot be explained by any appropriate durative verb and which even though it is not one of the broadest categories it epitomizes the meaning of many similar verbs. Nor do the verbs just treated take us to the previous large group of those denoting action. -- Let us make a try with our other process verb: felhősödik /clouds over/: válik /becomes/ → lesz 2 a /turns into sg/; of this the nearest synonym is válik /becomes/, thus we reach at once the synonymic domain of válik, lesz as the highest category. Here the most general verbal notion behind all the verbs in this group is perhaps expressed by változik /changes/. -- We need not make a separate exploration of the verb lesz since, as we have seen, felhősödik belongs to the category represented by the verb lesz. At best we can follow up keletkezik /arises/, another synonym of its meaning group I : létrejön /comes into being/ → válik → so once again we have come back to the same point.

As can be seen process verbs can really be traced back to such verbs of a more general meaning under whose concept verbs of this kind fall. If the analysis is carried out mechanically points of contact will be able to be established between action and process verbs, but it will be equally evident that these verbs constitute a rather closed group apart from a few links with the other group. What characterizes them as opposed to verbs of action is that they can have subjects other than persons /lesz/ or they cannot refer to an animate subject /felhősödik/. On the other hand, we have to separate process verbs from verbs of perception /hall/ because their subject is a person /occasionally animal/

but the process denoted by the verb does not depend on the agent's volition.

The explanatory verb <u>előidéz</u> /gives rise to/ of <u>csattan</u> /claps /once/, assigned to the category of occurrence, would lead to where the successive verbs of <u>ugat</u> /barks/ have led: to those denoting action with <u>tesz</u> /does, makes/ as the most general exponent of this set of synonyms. Therefore we have to take into consideration here the fact that meaning 1 of <u>csattan</u> refers to objects, things, consequently the next link in the explanatory string is meaning 2 of <u>létrehoz</u>, in accord with meaning 1 of <u>előidéz</u>. And that leads back to <u>előidéz</u>. It is in meaning 3 of <u>csattan</u> that we can find the verb that leads on to another verb category: <u>hangzik</u> /sounds, vi/. This, through the synonyms <u>hallatszik</u> /is heard/, <u>hallható</u> /audible/, brings out the passivity inherent in the meaning of <u>csattan</u>: it expresses not only that something happens to the subject, but also that we perceive it.

This verb leads on to the next large group, i.e. the verbs denoting various phenomena, where we get such synonym sequences as the following ones: <u>lobban</u>: "/lángot/ fog" /catches /fire/ → "keletkezik" /arises/ → ... "lesz", resp. "kicsap" /darts from/ → "előtör" /bursts forth/ → "árad" /streams/ → "terjed" /spreads/ → /lesz/ | <u>menny-dörög</u>: "hallatszik" → "hallható" | <u>villámlik</u>: /keletkezik/ → ... "lesz" | <u>csikorog</u>: "/hangot/ ad" /gives forth /noise/ → "létrehoz" → "előidéz".

The verbs belonging here, therefore, cannot be derived from identical verb categories through the string of explanations. Like <u>csattan</u> the verb <u>mennydörög</u> is to be qualified as denoting occurrence, while <u>lobban</u> and <u>villámlik</u> are similar to the process verb <u>felhősödik</u>. Standing apart, however, is the explanation of <u>csikorog</u> /creaks/, which leads over to the verbs denoting action of the <u>ugat</u>-type. The probable reason for this divergence is that the subject of <u>csikorog</u> is felt to be much more "active" than that of the verbs csattan and mennydörög both denoting similarly phenomena of sound. We can justify this feeling, on the one hand, by the fact that the verb csikordul /creaks /once/, differring only in temporal aspect, is likewise explained by the expression hangot ad /gives forth noise/, on the other hand, by the fact that in its meanings, except 1 d, we find a subject denoting thing, in fact, in meaning 2, man, animal, whereas the subject of csattan may be "sound", too, and mennydörög has no subject in its meaning 1. -- That the Dictionary ranges the meaning of csattan among "occurrences" in explaning the corresponding nomen actionis can be probably accounted for by the fact that its meaning is markedly momentary as opposed to "phenomena" which are usually perceptible for some time. The explanations of lobban, villámlik, on the other hand, are explained through the fact that we are more likely to apprehend visual perceptions as objektive change than those perceived by the ear.[8]

Let us now cast a look at the verb of motion folyik /flows/: halad /passes on vi./ —► jut /gets swhere/, from here, however, the explanatory string swerves away because from ér /arrives/ to eljut /gets swhere/ and from here further to előtünik /appears/ we find rather verbs of perfective aspect which together with "vhova" /swhere/ express the final point rather than the process itself of the motion. The other synonym of halad, közeledik /approaches/ takes us to the same point with a little detour: kerül /gets swhere /in some way/ —► jut /gets swhere /destination/. Since the string of the explanatory verbs does not lead to a verb of more general meaning, we can regard halad at once as the verb whose concept subsumes folyik as well as many other verbs. We can put the verb folyik in a small group of verbs which denote progress, progredient movement, that is, they constitute a well definable sub-group within the class of those denoting motion.

From the explanation of the state ver áll /stands/

the following set of verbs is obtainable: mozog /moves vi./ → van /is/ → tartozik /vhova/ /belongs /swhere/, resp. létezik /exists/. Mozog cannot be a superordinate synonym of áll as áll means "nem mozog" /does not move/. The verb van, however, represents a category which includes both mozog and nem mozog. The concept of existence subsumes both state and motion, but only in a very broad sense of the word, and therefore it seems expedient to treat all three as separate verb groups.

There is an even more general category of which we have quoted besides marad /remains/ the very verb van /is/. This is the category of "tény" /fact/. However, tény expresses something which includes the meaning content of any subject in conjunction with a finite verb /with transitive verbs often including that of the object, too/, and in the majority of nomina actionis the Dictionary does give it as a synonymous expression: "az a tény, hogy ..." /the fact that .../. E.g. in the introduction to the explanation of maradás: "Ált. a marad igével kifejezet magatartás, ill. állapot; az a tény, hogy vki, vmi marad" /Generally conduct and state resp. expressed by the verb marad; the fact that sy or sg remains/. In the detailed explanation of maradás or in explaining létel /existence/ the Dictionary does not give another genus proximum because these verbal nouns themselves express a general verbal content and can be subsumed immediately under the notion of "fact". Their underlying verb belongs to a distinct small group: the verbs denoting existence.

Finally we have to examine the group of verbs denoting sensation, relationship and manner. The explanatory string of the verb fáj /aches/ /"vmely testrészben ... gyötrő érzés jelentkezik" : in some part of the body a ... painful sensation appears/: jelentkezik → feltünik /appears/ → ... lesz → ... csinál, thus, on the one hand, it leads towards the verbs denoting change, process, on the other, towards those expressing causation. Anything that aches c a u s e s pain, in the course of which pain a r i s e s and we f e e l it at once. By this the sensation verbs of the fáj type come close to those denoting phenomena of the csikorog type. Even so it is useful to keep this group apart because the subjective feeling dominates in it. Pain can only appear in the consciousness, in the nervous system, while the noise called creaking may sound without the mind being aware of it.

The explanatory string of vonatkozik /relates to sg/ is this: van V 7 /is/ —→ tart /is suhere/ —→ folyik /flows/ —→ halad /passes on vi./ —→ közeledik /approaches/ —→ jut /gets swhere/, but here we are more and more on a wrong track because along this way we get to the concept of motion or even arrival rather than to that of relationship. The string of its other synonym is: összefügg /is in coherence with/ —→ áll /stands/ —→ van /is/. We thus reach the verb van in two ways and this verb usually requires to be complemented by another word into a phrase. This is rather characteristic of verbs expressing relationship: it is not they that refer directly to their subjects but show the relationship of the subject with some other nominal concept.

The verb tegeződik /address each other by thou/ is explained through the string: tegez /addresses sy as thou/, —→ szólít /addresses/ —→ él /2 [vmivel] avails himself of/ —→ alkalmaz /employs/ —→ használ /uses/ —→ vesz /takes/ —→ von /draws, pulls/ —→ kezd /begins/, but here we swerve away into the perfective direction of inchoation, therefore we have to remain within the scope of the synonyms of "uses", "employs", "avails himself of" in seeking the superordinate synonym of the verbs denoting relationship.

2.5. Let us now project the main verb categories obtained inductively from the explanations of some 300 verbs on to the categories of names of actions obtained from the explanations of the Explanatory Dictionary. With this largely mechanical, formal procedure we arrive at the following main verb groups and sub-groups after the most necessary rearrangement. /Each group is represented here by the highest level synonyms, occasionally complemented by other illustrative examples; the sub-groups make no pretension to completeness./

1. A c t i o n : tesz /does, makes/, végez /performs/, csinál /makes/, cselekszik /acts/, intéz /contrives, arranges, manages/, dolgozik /works/. Some characteristic sub-groups are as follows:

a/ Moving: mozgat /moves, vt/.

b/ Making arrangements: kifejez /expresses [his wish etc.]/, mond /tells/, intézkedik /sees to it/, gondoskodik /provides/.

c/ Establishing: létrehoz /bring into existence/.

d/ Manifestation: beszél /speaks/, szól /utters/.

e/ Producing sound: /hangot/ ad /gives forth [sound]/, előidéz /gives rise to/, hallat /lets sg be heard/.

f/ Indication: mutat /shows/, utal /refers/.

2. P r o c e s s : változik /changes vi./. Its characteristic sub-groups are:

a/ Change of state: válik /becomes/, /-bb/ lesz /becomes [more...]/. Here belong the verbs denoting change of state in a strict sense, e.g. elájul /faints/, elaprózódik /gets fragmented/, enged 1 /yields [to force]/ just as much as those denoting change of place, movement in a transferred sense, e.g. adatik /is [being] given/, kitudódik /gets known/, or those denoting change of quality, e.g. aljasodik /becomes depraved/, hül /gets cool/, melegszik /gets warm/.

b/ Origination: lesz /becomes/, keletkezik /originates/, létrejön /comes into being/.

c/ Momentum: kezd /begins/.

d/ Cessation: eloszlik /vanishes/, befejeződik /comes to an end/; here are also to be assigned abbamarad /is left off/, megszünik /ceases/.

e/ Perception: érzékel /perceives/.

f/ Participation: kap /receives/.

3. O c c u r r e n c e . Sometimes it shades off into the category of process. Some of the characteristic sub-groups:

a/ From the point of view of subjective perception: hangzik /sounds, vi/.

b/ From the point of view of the objective fact: történik /happens/, megtörténik /takes place/. Belonging here are also e.g. adódik /occurs/, elmarad /does not happen/.

4. P h e n o m e n o n . It is coterminous with the category of occurrence. Its sub-groups:

a/ Emphasizing the subjective viewpoint: hallatszik /is heard/, látszik /is visible/. Verbs of special meaning belonging here e.g. búg /booms/, ködlik /looms/, ragyog /shines/.

b/ Emphasizing the objective fact: ? előidéz /gives rise to/, ? keletkezik /arises/. E.g. ég /burns/, lángol /flames/.

5. S e n s a t i o n : ? keletkezik /arises/, ? jelentkezik /appears/. The verbs belonging here--e.g. fáj /aches/, viszket /itches/--overlap with those in the previous category.

6. S t a t e :

a/ Process-like motion: halad /passes on vi./, mozog /moves vi./, kerül /gets swhere/.

b/ Non-motion: áll /stands/, van /is/ /with an adverb of place or state/, marad /remains/. E.g. alszik /sleeps/, lebeg /floats/, megbujik /hides/.

c/ Mental State: ? eltölt /fills/. E.g. álmélkodik /marvels/, gyötrődik /suffers/.

d/ Position: vezet IV /leads/, vonul 3 /runs/, halad 3 /passes on vi./. E.g. aláhajlik /hangs down/, függ /hangs/, kanyarodik /bends vi./.

e/ Attitude: mutat /shows/, megnyilatkozik /manifests itself/, viselkedik /behaves/.

f/ Ability: tud /knows/, bír /is able/.

7. F a c t : van /is/, létezik /exists/, nincs /is not/, hiányzik /is missing/. Besides the pure verbs of existence, such as akad /there is occasionally/, előfordul /is, occurs/, those expressing necessity or possibility also belong here, e.g. kell /has to/, lehet /may be/.

8. R e l a t i o n s h i p : összefügg /is in coherence with/, tartozik I 1 /belongs/, vonatkozik /relates to/. E.g. alapul /is based on/, aránylik /is in proportion to/, egyezik /agrees with corresponds to/.

9. ?M a n n e r : használ /uses/, alkalmaz /employs/.

The items appearing with a question mark are non-substitutable, directly superordinated synonyms of the verbs assigned to their groups. Their syntactic patterns are different or they just overlap with the concept of the respective verb. The category of manner is problematic as a whole. The categories enumerated above are merely a consequence of the system of explaning verbs in the Explanatory Dictionary. This system needs correcting here and there.

2.6. Is there any theoretical basis for this classification based on the explanations, that is, mainly on intuitive linguistic competence? How would it be possible to characterize these partly distinct, partly overlapping verb categories?

These categories are primarily determined by the manner of relationship that obtains between the meaning content of the verb and its subject. There are, of course, exceptions, too, in which another subject takes over in the explanation; for example, of the explanatory verbs of process verbs <u>érzékel</u> /perceives/ and <u>kap</u> /receives/ stand apart in this respect and the same specific character attaches to <u>keletkezik</u> /arises/ and <u>jelentkezik</u> /appears/ out of the verbs of sensation, and to <u>előidéz</u> /gives rise to/ out of the verbs of phenomena. From this as well as other points of view the most homogeneous category is that of a c t i o n . It is characterized by the fact that the subject is the source, the starting point of the action, and in most cases, deliberately so. Since the majority of verbs belongs to this category, many grammar books use the label "action" to express the general meaning content of verbs /especially when contrasted with passive verbs/. The linguistic basis of this is that to find the verb of the sentence we usually ask the question <u>mit csinál</u>? /what does he, it, she do?/ The same anthropomorphic view plays a part also in other verb categories. /Cf. Manfred Sandmann: Substantiv, Adjektiv-Adverb und Verb als sprachliche Formen. Indog. Forsch. 1940, pp. 101-2; Karl Boost: Neue Untersuchungen zum Wesen und zur Struktur des deutschen Satzes. Berlin, 1955. p. 43./ Closest to human and animal activity stand those verbs which express the movement of vehicles of transport.

P r o c e s s is characterized by the fact that though it springs from the subject but not as from a cause and even less deliberately and furthermore that--apart from perception which stands closer to action-- the subject is not constant but changing or it may even be the result of the process itself. /The subjectless process verbs are state-like: <u>alkonyodik</u> "the sun is setting", etc./. -- O c c u r r e n c e as a narrower category differs from process in so far as the subject is very frequently an abstract rather than a concrete noun, further that it may express not only an objective concept of event but also the speaker's subjective impression. In a wider sense we could call occurrence the semantic content common to all verbs -- with a generalization more justified than using "action" for the same purpose. -- P h e n o m e n o n is the least homogeneous category. The subject here may be enduring and concrete /látszik vmely tárgy "some object is visible"/ or something transient /lobban a láng "the flame flashes /once/"/ or even the subject may be missing /mennydörög "it thunders", villámlik "it is lightning"/. The subjectless verbs mostly belong to this class. The presence or absence of the subject, however, is not a category-determining factor. Of much greater importance is the fact that the verb expresses the speaker's subjective impression too. Phenomenon is not a category primarily characteristic of verbs. The next few categories are even less so. -- In the category of s e n s a t i o n the subject of the verb may be likewise missing /fáj "it aches"/, but when it is present the verb does not express an objective fact but merely the speaker's subjective feeling. -- In the category of s t a t e the subjective impression is accidental or secondary, but, on the other hand, what is characteristic is that the relationship of subject and verb takes the most transparently simple form: pure existence or position or movement, that can be represented by a simple diagram or mechanical formula /áll "stands", halad "passes on", forog "rotates"/. S t a t e is even less characteristic of verbs than phenomenon is. /Of the subjectless verbs belonging here fagy "frost/ it freezes" is nomen as well as verbum!/ -- F a c t is also in reality a category not of the verb but rather of the objective content of the sentence as a whole. The Explanatory Dictionary employs this concept in itself when the verb itself to be explained stands for some general category. -- For the expression of r e l a t i o n s h i p another nominal concept is absolutely necessary besides the subject of the verb. -- M a n n e r is also not a genuine verb category. The

Explanatory Dictionary explains with its help such nomina actionis as are characterized by the manner of a verb category /ususally that of action/.

Proceeding quite mechanically in tracking down the verbs explaining others in the Dictionary would often put us, as we have seen, on a false scent unless we do not fail to take into account the specific relationship that exists between the verb explained and its subject as well as another important factor, namely that the precise meaning of a verb is sometimes equivalent not to that of another verb but to an expression or group of words /and often one in which the verb belongs to a very different category/. Thus we may conclude that verbs are characterized by not only their subjects but also by what nouns of a different grammatical form they enter into frequent or necessary combination especially of a kind which may have a meaning quite different from that of the simple verb.

NOTES

[1]"A sound sentence-semantics is only conceivable on the basis of a well-elaborated word-semantics." Ferenc Kiefer: A Katz—Fodor--féle szemantikaelmélet /Katz—Fodor's Semantic Theory/, MTA I. O.K. /Proceedings of the Division of Linguistics and Literary Studies of the Hungarian Academy of Sciences/,Vol. 23 /1966/, p. 187.

Section 1 of the present paper, in a more elaborate form, was read at a meeting of the linguistic group of the Computational Centre of the Hungarian Academy of Sciences on 22nd May, 1964. Section 2 has been published in Hungarian under the title "Az igék szótári ábrázolásáról" /The lexicographic treatment of Hungarian verbs/, Chap. II, Szótártani Tanulmányok /Studies in Lexicography/, ed. by László Országh, Budapest, 1966, pp. 184-194. The lecture was reviewed and the ideas developed further by F. Kiefer in Általános Nyelvészeti Tanulmányok /Studies in General Linguistics/, Vol. 4 /1966/, pp. 133-140.

[2] Cp. A szótárírás elmélete és gyakorlata a magyar nyelv Értelmező Szótárában / Theory and Practice of Lexicography in the Explanatory Dictionary of the Hungarian Language/, ed. by László Országh, Budapest, 1962, pp. 68-70.

[3] On the two kinds--broader and narrower--synonymity see Gábor O. Nagy: A rokonértelműség szinonimaszótári szempontból / Synonymity from the Synonym Dictionary Point of View/, Studies in Lexicography, 114-117 and László Elekfi, ibid. 186.

[4] The English translation does not perforce always render exactly the semantic structure of the Hungarian illustrative examples. The strings would naturally work out differently in English. The bracketed English equivalents are therefore tentative only.

[5] These strings may be represented by graphs. There is no room here, however, to consider the relevant aspects of the graph theory. The question was dealt with by László Kalmár in one of his earlier lectures to the Hungarian Linguistic Society /November 3, 1961/; see further Általános Nyelvészeti Tanulmányok /Studies in General Linguistics/, Vol. 2 /1964/, pp. 45-46. -- Theoretically each member of an explanatory string stands with the next under it in the relationship of inclusion. Cf. Katz, J.J. and Fodor J.A.: The Structure of a Semantic Theory, Language 39 /1963/, pp. 189-192; F. Kiefer: Proceedings, etc. /1966/ p. 179.

[6]The term "synonym" is used throughout this section in its broadest possible sense, including strings of words in apposition or in coordinate series as well as the generic term designating a concept higher than that for which the entry word stands. See also footnote 3 above.

[7]His stands in the translations for her and its as well.

[8]Edit Vértes exemplifies Brøndal's sub-group of "expressive verbs" by kopog /knocks/: Általános Nyelvészeti Tanulmányok Vol. I. /1963/, p. 332.

MEANING, SYNONYMY AND TRANSLATION

L. Kalmár

The aim of the present paper is to contribute to the development of a formal semantic theory with the publication of some initial results of research dealing with the logical interrelations among the notions of meaning, synonymy and translation. It also proposes to outline the work to be done.

At first sight the question of logical interrelations among the three notions mentioned above seems trivial. Two units belonging to some linguistic level /e.g. two morphemes or two sentences/ are synonymous if and only if their meaning is the same. The meaning of a linguistic unit is that something which is common in the units synonymous with it /including also itself/, or applying the usual set theory refinement of such definitons based on abstraction the meaning of a unit is that abstraction class with regard to synonymy as an equivalence relation which includes the form in question, or generalizing this method of set theory refinement, the meaning of a unit is the object assigned to this abstraction class where we can choose at will those objects which we assign to the abstraction classes belonging to the synonym with the only condition that the assignement should establish a one-to-one correspondence. Finally, translation is such a mapping of the units of a certain linguistic level of the source language onto the units of the target language belonging to the same level that leaves the meaning of the units unchanged.

But we have such simple relations only if homonymy is ignored. This can usually be justified methodically by the assumption that the elimination of homonymy /e.g. the addition of a homonymy-index to morphemes, or that of symbols expressing the results of the syntactic analysis of sentences, e.g. the provision of P-markers/ must precede semantic analysis. In practice, however, when eliminating homonymy we have to rely on semantic notions, first of all the notion of meaning. Consequently, we think it more correct to study the logical interdependence of the notions synonymy, meaning and translation with regard to the phenomenon of homonymy, even when this needs an examination of more complex interrelations than those outlined above. Also we have to eliminate the logical jump involved in the above reasoning which tries to define the notion of meaning within one language while in the definition of translation it already regards it as an interlinguistic notion.

Az I have done so far in this paper I am not going to state definitely on which linguistic level I am examining the three mentioned semantic notions. Although it is quite unusual to speak about translation on a lower level than the level of the sentence, in reality we always translate a text of the source language into the target language, and the sentence by sentence translation is only an approximation to the text translation taken in the strict sense, even though it is a better approximation than the word by word /or morpheme by morpheme/ translation. We do not usually call a grapheme by grapheme rendering a translation /that is, an approximation to the text translation/ but a transliteration.

For the sake of simplicity, however, let us remain within the scope of one single language, at least for the time being, and even within this frame at a definite linguistic level, and let us try to clarify the logical interrelations between the notions of meaning and synonymy taking homonymy into due account.

From a mathematical point of view the situation is the following. There are two sets, a set S of certain forms belonging to the language in question /in general complex symbols, markers/, furthermore a set Ω of certain objects /denoted things/, which can be denoted by the elements of S. Furthermore we consider a subset D of the Cartesian product of the sets S and Ω which include those and only those ordered pairs (s, ω), where s is some element of the set S, while ω is such an element of set Ω that can be denoted by the linguistic form that is such an object which is a possible meaning of the form s. A form s $(s \in S)$ can have more than one meaning: several objects ω $(\omega \in \Omega)$ can have $(s, \omega) \in$ D /homonymy/ and vice versa, an object ω can be denoted by several form s /synonymy/.

Two forms, s_1 and s_2, may be called weak /or partial/ synonyms, if they have at least one meaning in common.

$$s_1 \sigma s_2 \leftrightarrow \exists \omega((s_1, \omega) \in D \wedge (s_2 \omega) \in D), \qquad /1/$$

where σ denotes weak synonymy and the variables ω /with or without index/ range over the set Ω /while the variables s with or without index over the set S/.

Weak synonymy thus defined is evidently a symmetrical relation:

$$\forall s_1 \ \forall s_2 (s_1 \sigma s_2 \longleftrightarrow s_2 \sigma s_1).$$

If we assume it as an axiom that every form has at least one meaning:

$$\forall s \exists \omega((s, \omega) \in D). \qquad /2/$$

that is, we leave out of S the meaningless forms then weak synonymy becomes a reflexive relation:

$$\forall s (s \sigma s).$$

/The meaningless forms would not be weakly synonymous between themselves./ With an example we can easily show that weak synonymy is not a transitive relation.

The question of the definition of meaning with the help of the notion of synonymy, if, for the time being, we interpret synonymy as weak synonymy, will lead to the following mathematical problem: Be given the set S and the reflexive and symmetrical relation σ defined in it. Is it possible to find such a set Ω that by suitably choosing the subset D of the Cartesian product $S \times \Omega(1)$ is fulfilled? To what extent does the relation determine the set Ω disregarding, of course, the notations of their elements, and also the set D?

I am going to demonstrate that the sets Ω and D can always be chosen in the specificd way, and generally in more than one way. For this purpose it is convenient to represent the σ relation with a graph, the verteces of which represent the elements of the set S, Furthermore, two of its verteces are then and only then connected by an edge if the relation holds among the elements of S represented by then. We denote this graph also with σ , and its verteces in the same way as the element of S represented by them. Since σ is a reflexive relation, from each of the verteces of the graph a " loop line " leads into itself. We leave these loops out of graph σ but even without them each of its verteces is considered to be connected with itself.

If such sets Ω and D exist, then for any of the elements Ω of ω the following is valid: those verteces of the graph to which /that is to the elements of the set S represented by them/ $(s, \omega) \in$ D holds are linked together in pairs because these elements are weakly synonymous for ω is their common meaning. Consequently all such verteces are the verteces of a total graph σ_ω which is the subgraph of the graph σ .

Any of the verteces s of the graph σ is also a vertex of at least one such total subgraph σ_ω because according to the axiom /2/ there is at least one such element ω of Ω for which $(s, \omega) \in$ D. Furthermore, any edge of the graph σ is the edge of at least one total subgraph σ_α . If, namely, the edge in question connects the verteces s_1 and s_2 of the graph σ then the s_1 and s_2 elements of the set S are weakly synonymous, that is Ω has at least one such element ω for which $(s_1, \omega) \in$ D as well as $(s_2, \omega) \in$ D. In this case both s_1 and s_2 are verteces of the subgraph σ_ω, consequently, as σ_ω is a total graph, the edge connecting the two verteces is an edge of σ_ω.

Conversely, let be given such a set M of the subgraphs of the graph $\mathfrak{G}$, that a/ each graph belonging to M is a total graph, furthermore, b/ each vertex of the graph $\mathfrak{G}$ is also a vertex of at least one graph belonging to M, and c/ each edge of the graph $\mathfrak{G}$ is at the same time an edge of at least one graph belonging to M, then we can obtain the sets Ω and D of the required property in the following way. Let us assign to the graphs belonging to M an arbitrary object in a one-to-one correspondence; be Ω the set of these objects /e.g. Ω be M itself, if we assign to each graph belonging to M, the graph itself./ Furthermore, let D be the set of all those ordered pairs (s, ω) , where $s \in S$, $\omega \in \Omega$ and which fulfil the condition that the vertex s of the graph $\mathfrak{G}$ is at the same time the vertex of the graph belonging to M to which the object ω is assigned. In this case /1/ is fulfilled. Indeed, be s_1 and s_2 two such elements of S that fulfil the relation $s_1 \mathfrak{G} s_2$, that is the verteces s_1 and s_2 of the graph $\mathfrak{G}$ are connected by an edge. This edge is because of c/ the edge of some graph belonging to M, let ω be the object belonging to this graph. Then $(s_1 \omega) \in$ D as well as $(s_2, \omega) \in$ D, because both s_1 and s_2 are verteces of that graph belonging to M to which the object ω is assigned. Vica versa, if for some object ω belonging to Ω $(s_1, \omega) \in$ D, and $(s_2, \omega) \in$ D, i.e. both of the verteces s_1, s_2 of the graph $\mathfrak{G}$ are at the same time verteces of that graph belonging to M to which the object ω is assigned then because of a/ s_1 and s_2 are connected with an edge within this graph, consequently, since this graph is part of graph $\mathfrak{G}$, this edge is also an edge of $\mathfrak{G}$, i.e. the relation $s_1 \mathfrak{G} s_2$ will be fulfilled. Az a consequence of condition b/ also the axiom /2/ will be fulfilled.

Thus we have yet to show that the subgraphs of the graph $\mathfrak{G}$ have such a set M which satisfies a/, b/ and c/ conditions. Such a set M is formed by the isolated verteces that is, those not connected with any other vertex of the graph $\mathfrak{G}$, as total graphs with one single vertex, further the edges of the graph $\mathfrak{G}$, as total graphs of two verteces. But such a set M form also the maximal total subgraphs of the graph $\mathfrak{G}$, that is the total subgraphs of $\mathfrak{G}$ such, that not a single vertex of the graph $\mathfrak{G}$ is connected by means of an edge of the graph $\mathfrak{G}^*$ with each vertex of the graph $\mathfrak{G}$, unless the vertex in question is not a vertex of the graph $\mathfrak{G}^*$, /Each total subgraph of the graph $\mathfrak{G}$ is the element of some maximal total subgraph of $\mathfrak{G}^*$. Thus each vertex of $\mathfrak{G}$ can be interpreted as a total subgraph with one vertex, and all its edges as total subgraphs with two verteces. That means that each of the verteces of $\mathfrak{G}$ is also the vertex of some of its maximal total subgraphs and each of its edges is also the edge of some of its maximal total subgraphs. We can obtain such a maximal total subgraph from the total subgraph $\mathfrak{G}'$ of the graph $\mathfrak{G}$ by adding to it a vertex /if there are more we can choose one at will/ of the graph $\mathfrak{G}$ which is connected by means of an edge of the graph $\mathfrak{G}$ with any one from among its original or added verteces but we add also the connecting edges until the graph has some left.

The two sets M that we have mentioned by way of illustration are in general different /disregarding the trivial case when graph $\mathfrak{G}$ has no total subgraph with three verteces which would correspond to a language that does not exhibit three, pairwise weakly synonymous forms. This shows that weak synonymy taken in itself is not suitable for the definition of the meaning of linguistic form, not even if we leave out of consideration the choice of the objects acting as meaning. Hence, if we want to define the notion of meaning with the help of synonymy then, beside weak synonymy, we have to take into consideration also some other kinds of relations of synonymy as well.

Such a relation is, first of all, strong synonymy. We call two linguistic forms s_1 and s_2 strongly /or totally/ synonymous if all their meanings are common:

$$s_1 \Sigma s_2 \leftrightarrow \forall \omega ((s_1, \omega) \in D \leftrightarrow (s_2, \omega) \in D)$$

/3/ where denotes strong synonymy. In other words, two forms are strongly synonymous if and only if the sets of their meanings are equal:

$$s_1 \Sigma s_2 \leftrightarrow \hat{\omega}((s_1, \omega) \in D) = \hat{\omega}((s_2, \omega) \in D).$$

From this it directly follows that strong synonymy is a reflexive, symmetrical and transitive relation, i.e., it is an equivalence relation:

$$\forall s (s \Sigma s),$$

$$\forall s_1 \forall s_2 (s_1 \Sigma s_2 \leftrightarrow s_2 \Sigma s_1),$$

$$\forall s_1 \forall s_2 \forall s_3 ((s_1 \Sigma s_2 \wedge s_2 \Sigma s_3) \rightarrow s_1 \Sigma s_3).$$

Furthermore, if two forms are strongly synonymous then they are also weakly synonymous:

$$\forall s_1 \forall s_2 (s_1 \Sigma s_2 \rightarrow s_1 \sigma s_2), \qquad /4/$$

for if forms coincide in every meaning then they have a common meaning since according to /2/ they have meaning. Finally, if among three forms the first two are weakly and the second and the third strongly synonymous then the first and the third are weakly synonymous:

$$\forall s_1 \forall s_2 \forall s_3 ((s_1 \sigma s_2 \wedge s_2 \Sigma s_3) \rightarrow s_1 \sigma s_3), \; /5/$$

because in this case the first two forms have common meaning and this is the common meaning also of the first and the third forms for each meaning of the second form is the meaning of the third form as well.

The question of the definition of the notion of meaning with the help of the notions of weak and strong synonymy leads to the following mathematical question: Be given the set S, and the reflexive and symmetrical relation σ defined in S, finally an equivalence relation Σ interpreted in S which additionally fulfill also conditions /4/ and /5/. Is it possible to find a set Ω and a subset D of the Cartesian-product $S \times \Omega$ such that assuming axion /2/ /1/ and /3/ are fulfilled? To what extent do the relations σ and Σ determine the set Ω /disregarding the notations of its elements/ and the subset D of the Cartesian-product $S \times \Omega$?

With a reasoning similar to the above, it can be demonstrated that the sets Ω and D can always be chosen in the required way. It is again expedient to visualize relations σ and Σ by means of a graph as explained in the above; these graphs will be denoted by σ and Σ , resp. and their verteces will be denoted in the same way again as the elements of the set S represented by them. The graph can be divided into such /uniquely determined, maximal/ total subgraphs, which have no common vertex /not even two of them have/ because Σ is an equivalence relation. This graph Σ is, because of /4/, the subgraph of graph σ . Further, if, as a consequence of /5/, a vertex of Σ is connected by an edge σ with a vertex of the maximal total subgraph of σ , then it is connected by means of an edge of σ with each vertex of this maximal total subgraph. This circumstance makes it possible to define the so-called vector graph of σ with respect to the subgraph of Σ . We obtain this factor graph from σ by replacing each of those verteces with a new one, that are verteces of the same maximal total subgraph of Σ , and we connect two such new verteces with an edge if and only if each /as pointed out above, any two/ of the verteces of those maximal total subgraphs of Σ whose verteces have been replaced by the new verteces in question, are connected by an edge of σ . This factor graph is usually denoted by σ / Σ .

According to a reasoning similar to the above we obtain all possible sets Ω chosen in the required fashion and the sets D belonging to them in the following way. Consider a set M of the subgraphs of the graph σ/Σ that fulfills conditions a/, d/ and c/ in which, however, the graph σ is to be replaced by the factor graph σ/Σ . We establish a one-one correspondence between the graphs of the set M, and some objects. The set of these will be Ω . D will be the set of those ordered pairs (s, ω) for which $s \in S$, $\omega \in \Omega$ and which satisfy the condition that the vertex of the factor graph $\mathfrak{G}/\Sigma$ that replaces the s vertex of the graph $\mathfrak{G}$ in this factor graph and also those verteces of $\mathfrak{G}$, that are verteces of the maximal total subgraph that contains also s among its verteces, is also the vertex of that graph of M to which the object ω has been assigned.

If these Ω and D sets were uniquely defined then the meaning of the forms belonging to M could be defined with the help of the notions of weak and strong synonymy, as follows. The meaning of a form s is the object assigned to the graph belonging to the set M to the verteces of which belongs the vertex of the factor graph $\mathfrak{G}/\Sigma$ that replaces the vertex s of the graph $\mathfrak{G}$ in this factor graph /among others/ where $\mathfrak{G}$ and Σ are the graphs depicting weak and strong synonymy in the above manner, while M is the set of the subgraphs of the factor graph $\mathfrak{G}/\Sigma$ satisfying the above a/, b/ and c/ conditions, though in the conditions b/ and c/ the graph $\mathfrak{G}$ should be replaced by the factor graph $\mathfrak{G}/\Sigma$. Finally, the assignement of the objects to the graphs of the set M brings about a one-to-one correspondence.

However, the set M, except for the trivial case when the factor graph $\mathfrak{G}/\Sigma$ has no total subgraph with three verteces, can be chosen in several ways. Consequently, the notions of weak and strong synonymy are not even together suitable to define the notion of meaning with their help. For this purpose we have to take into consideration some further notions of synonymy.

A great number of synonymy relations can exist between two linguistic forms, e.g. they may have at least two meanings in common, or may have at least three common meanings, etc., with one exception at most each of their meanings is common /that is, they have the same number of meanings and among them there is only one different, and not more, or each of the meanings of a form is also the meaning of the other form, but this later has one additional meaning/, all their meanings are common except maximum two, etc., furthermore, there may be such a relation where two forms have more meanings in common than different ones. It seems probable that generally it is possible to define precisely the notion 'synonymy relation' with taking into consideration the structure of the formula expressed in terms of the symbols of mathematical logic which formalizes the definition of 'relation'. /See formulae /1/ and /3/./ Of course, these synonymy relations, too, reveal certain characteristics. /Namely, weak synonymy is reflexive and symmetrical, strong synonymy is an equivalence relation/. On the other hand, there are also certain connections among them /as, for example, the connections between weak and strong synonymy expressed by formulae /4/ and /5/./

However, we need not consider every possible synonymy relation. It would be enough to find a complete system of synonymy relations in the sense that the synonymy relations belonging to this system, except for the one-to-one correspondence, uniquely define the meaning of the linguistic forms. More precisely, the formulae (δ) defining the synonymy relations belonging to the system together with the formulae (ε) formalizing the characteristics of the synonymy relations belonging to the system and also, the synonymy relations belonging to relations existing between these characteristics have the property that it is always possible to find a set such that for the given set S and the relations interpreted on it, that correspond to the formulae (ε), leaving out of consideration the notation i.e., the one-to-one correspondence, in a unique way, furthermore, it is possible to find in a unique way, a subset D of the Cartesian-product $S \times \Omega$ such that beside axiom /2/ the formulae (δ) are fulfilled.

If we knew a total system of synonymy relations then with the help of the synonymy relations belonging to this system, or with the help of the graphs depicting them /which is the same/ we could define the meaning of the linguistic forms, but, of course, in a more complicated way, than with the aid of the abovementioned notions of weak and strong synonymy used for the definition of meaning /which is unsatisfactory because these relations do not form a total system/.

However, the question, or better to say, the problem of how to render a total system of synonymy relations, is, for the time being, mathematically unsolved. At present I cannot even prove that such a system exists, although this seems highly probable.

4. As I have already mentioned, no definition of meaning can be regarded as satisfactory from the viewpoint of the theory of translation if it tries to solve this problem within the frame of one single language and thus, does not consider also the logical, interlinguistic character of meaning. An entirely satisfactory definition of meaning must be based on the corresponding interlinguistic notion, on the notion of translation and not on that of synonymy within one language.

Let us confine ourselves to only the simplest "interlinguistic case", the case of two languages. Remaining at one single linguistic level, we consider three sets, the set S_1 of forms belonging to the level in question of one of the languages, the set S_2 of the forms belonging to the same level of the other language, and lastly the common set Ω of the meanings of the forms. Furthermore we consider a subset D_1 of the Cartesion-product $S_1 x \Omega$, and a subset D_2 of the Cartesian-product $S_2 x \Omega$: D_1 contains the ordered pairs (s_1, ω) , and only those, for which $s_1 \in S_1, \omega \in \Omega$ and ω is the /or a possible/ meaning of the form s_1, while D_2 contains those, and only those ordered pairs (s_2, ω) for which $s_2 \in S_2$, $\omega \in \Omega$ and ω is the /or a possible meaning of the form s_2. Let us assume once again, that each of the forms belonging to S_1 or S_2 has at least one meaning.

We call the form s_2 belonging to the set S_2 to be the weak translation of the form s_1 belonging to the set S_1, if the forms s_1 and s_2 have at least one meaning in common, that is, there is at least one such object for which

$$(s_1, \omega) \leftarrow D \wedge (s_2, \omega) \in D$$

holds. We say that s_2 is the strong translation of s_1, if every meaning of s_1 and s_2 is common, that is, if

$$\forall \omega ((s_1, \omega) \in D \leftrightarrow (s_2, \omega) \in D).$$

We can define in a similar way further translation notions falling between the notions of weak and strong translation as well.

In order to arrive at a definition which takes into account also the proper, interlinguistic character of meaning, we have to study the characteristics of the translation notions as well as the relations holding between them, if necessary with the assumption of some further axioms. One such axiom could be that any of the forms of a language has at least one translation possibility in the other language and vice versa, each form of the second language is the translation of a form of the first. Or, we could assume it as an axiom within one single language i.e. that any object ω has at least one such form, which is the only meaning of ω . /"That which can be expressed at all, can uniquely be expressed"/ Finally, a total system of translation notions should be found, in the same sense as I have explained in tha above in connection with the total system of synonymy relations, and we should define the notion of meaning with the help of the translation notions belonging to this system.

Such a definition could also show what kind of a meaning notion should we apply in order to satisfy the given requirements in connection with translation. This is essential because even on the same linguistic level we could speak about several weak or strong translation notions because we can raise different requirements as to the kind of nuances in meaning the translation should express. On the other hand, we can relax the requirement that the forms of the source language should have a grammatical translation in the target language, the only important thing being that the meaning of the form in the target language should be comprehensible. Thus, studies concerning the degrees of grammaticality could also influence the theory of translation.

THE POSSIBILITY AND/OR NECESSITY OF CS-RULES IN CATEGORIAL GRAMMAR[1]

Part 1

F. Kiefer

1. One might think of three different problems that can adequately be handled by means of CS-rules but only very clumsily, if at all, by CF-rules. The first one refers to the lexicon and to the assignment of categories, or category labels. Let us consider a word stem x and its paradigm $x_1, x_2, \ldots$. It is hardly conceivable to define x as the intersection of all x_i' s because there are very often slight or drastical changes in the stem according to the attached endings. A change might be caused simply by a svarabhakti vowel, as in Swedish "vinter-vintrar", by gender agreement "god-gott", by umlaut as in "bok-böcker", "lang-längre" etc. or by ablaut as in "sitta-satt-suttit", finna-fann-funnit" etc. Now it is partly a question of economy whether we want separate entries for each of the "stems" or whether we would rather like to have rules that account for the difference in the stem of the same word. It is, however, not only a question of economy, for if we do not account for the morphonology of a language, it might be necessary to list in the lexicon arbitrary phoneme-combinations like "vint" as though they were morphemes. A preliminary microgrammar for Swedish adjectibes by Hans Karlgren and Bengt Svensson shows clearly this point[2]. Thus by listing each variant of x we may run the risk of making counterintuitive ad hoc decisions. In the next section we shall investigate whether this could be avoided by setting up appropriate CS-rules.

The second problem we want to tackle is in close connection to the first one, though of quite different nature. If a word y can stand in n different positions in grammatical sequences, then the number of category labels to be assigned to y must be multiplied by n. It may also happen that n is infinite by recursive embeddings. If n is finite, the problem is solvable within categorial grammar, but if not, then another device is needed.

Thirdly, it is quite clear that discontinuity exceeds the power of CF-grammars. All the solutions that can be figured out in order to determine the syntactic relationship between discontinuous constituents amount to CS-grammar - type rules.

2. First let us turn to the svarabhakti vowel. The tentative rule to be given below is supposed to cover the following cases:

cykel-cyklar	seger-segrar	öken-öknar
segel-seglet	lager-lagret	tecken-tecknet

The e in the last syllable must be unaccentuated. That means, we have to assume the presence of at least one accentuated syllable. On the other hand, the dropping of e occurs before l, n, r, but not before other consonants. The dropping is effectuated if an ending beginning with a vowel has been attached to the stem. In view of the aim of this work we will not make use of the distinctive feature notation here. For the purpose of analysis, letters rather than phonemes should be considered as being given. Context-sensitive rules are denoted by double arrows. The sign # stands for word boundary, comma for option, parentheses for another kind of option (this will be clear from the rule). The rule is written like a generative rule and, as a matter of fact, can also be interpreted as being one, but it is rather the reverse of a generative rule if the whole set of rules for

Swedish were to be considered. Now the rule may be given in the following form for substantives:

(1) $$\underbrace{\text{Syllable Cons}}_{A}\quad \underbrace{\{l,n,r\}\ \{a,e\}\ \text{Cons}\ \{s,nas\}}_{B}\#\Longrightarrow A\ e\ B\ \#$$

We need a separate rule for adjectives because of cases like

enkel	vacker	vaken
enkelt	vackert	vaket
enkla	vackra	vakna
enkle	vackre	vakne[3]

As is readily seen we need even two rules, one covering the first two types and one extra rule for the "vaken"-type adjectives.

(2) $$\underset{A}{\text{Cons}}\quad \underbrace{\{l,\ r\}\quad \{a,\ e\}\#}_{B}\Longrightarrow A\ e\ B\ \#$$

and

(3) $$\text{Cons}\quad n\quad \{a,\ e\}\#\Longrightarrow \text{Cons}\quad e\quad n\ \#$$

Notice, incidentally, that (1) covers eight rules, (2) four rules. Thus we need 13 rules altogether in order to account for the svarabhakti vowel in the case of substantives and adjectives. If we can find a suitable way to build in these rules into the algorithms based on categorial grammar, then it would certainly be worth while to do so, because it would considerably reduce the number of different lexical entries. We will take up this problem in a later section.

In contrast to the svarabhakti e, umlaut is quite irregular and also unproductive. It is, of course, always possible to find some ad hoc description of which a simple list is often the best solution . One might still think of other possibilities. In view of the 20 and odd substantives that form their plural by adding an ending (in most cases) and by umlaut, it is possible to give a rule like

$$(4) \qquad \underbrace{(w)\ \text{Cons-cl}}_{A}\ \{o,\ a\}\ \underbrace{\text{Cons-cl}\ \{e,\ er\}}_{B} \Rightarrow A\ \{ö,\ ä\}\ B$$

where w stands for a word (morpheme) like in "ledamot", "bokstav" (maybe the only instance where this is the case), Cons-cl denotes a consonant cluster (consisting of one, two, or three consonants).

However, before the rule (4) is applied the stem of the word must be looked up in the dictionary without alteration. At the same time, however, one should also look for the umlauted form in the lexicon. The example "land-länder" makes the latter point clear. This above condition corresponds to the principle that in a generative grammar the lexical characterization of the words that are umlauted must contain a marker, let us say, +Omljud, to indicate this fact.

A similar rule could be set up for the adjectives for comparatives and superlatives and maybe also for irregular verbs. These rules will anyhow be of a somewhat ad hoc character but here it lies in the nature of the Swedish language itself.

3. Next we shall consider the forming of the neutral forms of the adjectives out of the forms in common gender. The latter are generally used as dictionary entries because they are shorter in most cases and/or represent the stem of the adjective. What ending the adjective in question will get depends solely on the form of the adjective, i.e. mostly in which kind of sequence it ends. Therefore, even disregarding the problem of concordance, one would get context-sensitive

rules. If we consider the forming of neuter adjectives as a separate problem, we have to put up lexical rules, i.e., rules operating in the lexicon. These rules would not explain anything about the syntactic behaviour of the adjectives but they might be necessary for reason of economy. Here they are:

(5)

$$\begin{aligned}
&+\text{Cons}\ \# \longrightarrow +\text{Cons t}\ \# \\
&+\text{Vowel}\ \# \longrightarrow +\text{Vowel tt}\ \# \\
&+\text{Syll ent}\ \# \longrightarrow +\text{Syll et}\ \# \\
&\text{dt}\ \# \longrightarrow \text{tt}\ \# \\
&\{+\text{Cons, a}\}\ \text{tt}\ \# \longrightarrow \{+\text{Cons, a}\}\ \text{t}\ \#
\end{aligned}$$

These rules are to be applied in sequence.

Now if we want to make use of the minimal syntactic information that is needed to account for (5) we must have additional rules that look somewhat like

(6) $Art_n \quad Adj \quad N_n \Longrightarrow Art_n \quad Adj_n \quad N_n$

and

(7) $Art_t \quad Adj \quad N_t \Longrightarrow Art_t \quad Adj_t \quad N_t$

where the subscripts $_{n,\ t}$ indicate n- and t-gender.

In syntax it seems reasonable to utilize an adjective "without gender" first and then attach gender (and the corresponding endings) by means of context-sensitive rules. (6) will, in view of (5), be empty. Once again, a more careful formulation could avoid this gap. It should also be made clear that (6) and (7) are not satisfactory from a syntactic point of view either because they should rather be part of a general agreement rule between adjectives and nouns, including plurals and definite forms as well. The fact that phonological rules operate on surface structures accounts for the fact that we have structures like Art Adj N, regardless of whether they have been generated from a separate sentence or not.

From the point of view of recognition it should be borne in mind that rules like (6) and (7) already require a sort of syntactic information in order to operate properly. Therefore, the placement of these rules in the system of recognition rules must not be done hastily.

As can readily be seen there is an essential difference between the svarabhakti rules, (5), and (6), (7), respectively. While the former can never be given in terms of categories whatever grammar one might happen to choose, the latter are categorial rules in the broad sense of this term. Thus, (6) and (7) could very easily be reformulated in terms of categorial grammar.

What is the status of (5) in the grammar? It would seem that the role of (5) is twofold. On the one hand, these rules reduce the number of lexical entries considerably, i.e. they are needed for the purpose of economy. On the other hand, however, with their help one can avoid assigning categories to non-morphemes like "go" for "god" and "gott", "e" for "en" and "ett" etc. Consequently, we may claim with good reason that these rules are necessary in grammar and not only possible. In other words, CF-rules are not enough in order to account for morphonological phenomena.

By observing that in (6) and (7) the article is dominated by the gender of the noun and by reversing the order we may reformulate these rules as (8):

(8) $Art_n \quad Adj_n \quad N_n \Longrightarrow Art \quad Adj \quad N_n$

$Art_t \quad Adj_t \quad N_t \Longrightarrow Art \quad Adj \quad N_t$

For the article we might have the rule

(9) $e \quad X \Longrightarrow en$ /before N_n

$e \quad X \Longrightarrow ett$ /before N_t

X stands for the morphoneme of "n" and "tt". Then, (8) may be changed accordingly.

In Swedish it might be possible to determine the noun phrases in a sentence without taking into consideration gender or agreement. If so, then the agreement rules should only operate after the noun phrases have been spotted. Another consequence would then be that we would not need to reverse agreement rules. More precisely, suppose that our noun phrase consists of an article, an adjective and a noun. The gender of the noun is identified but nothing else of this kind for adjectives and articles. Then by utilizing the context-sensitive generative agreement rules and lexical rules like (9) , one may arrive at the rught-hand side of one of (8), which is checked against the real sequence. This method is essentially the so-called analysis by synthesis approach which could be very simple in this particular case.

4. On the syntactic level there seem to be three types of cases where the use of context-sensitive rules is advisable or even necessary. One is the discontinuity problem. Let us consider the following sequence

(10) $x_1, x_2, \ldots, x_i, \ldots, x_j, \ldots, x_{n-1}, x_n$

Let us assume that (10) should be analyzed by a categorial grammar. Then to each x_k some categories (one or more) will be assigned. The categories will be realized as category labels. The grammar contains but two cancellation rules.

Let us now assume that x_2 and x_j belong together and x_i and x_{n-1} belong together. Thus, there are not only discontinuous constituents in (10) but we also have non-projectivity. Any non ad hoc solution will involve permutation. The "Fernwirkungen" described by Hans Karlgren[4] and me are hidden permutation rules. If x_2 and x_j form a syntagm this amounts to saying that the corresponding category labels must be reducible. The same holds, of course, for x_i and x_{n-1}. No cancellation rule can, however, be given in the form

(11) $x/y \quad z \quad y \longrightarrow x \quad z \qquad \text{or} \quad z \quad x$

where z stands for one or more category labels. Regardless of the conditions that must be imposed on "Fernwirkung" which are anything but clear to me .

(11) ought to be reduced to

(12) $x/y \quad y \quad z \longrightarrow x \quad z$

where z is not even necessary and the ambiguity of (11) does not arise. In order to get (12) out of (11) we must have the rule

$$AB \longrightarrow BA$$

in our grammar with some further specifications and/or restrictions.

Whenever we state the conditions for a cancellation, we have in fact a CS-rule. Thus, context-sensitive recognition seems to be a necessity, not only a possibility. And the necessity goes far beyond economical necessity. Of course, to write such a recognition algorithm is a task in itself. Let me make clear, however, that context-sensitive rules can as well be part of a categorial grammar as they can of a phrase structure grammar. Formally, the difference would be here that the conditions must be stated in terms of categories or category labels, respectively.

5. Certain morphemes (mostly adverbs, I presume) cause difficulties because they can move rather freely in the sentence and can modify various other types of morphemes. If A is such a morpheme and x_1, x_2, x_3 are categories, then the

following example shows that to have context-sensitive rules does not necessarily mean a simplification.

(13) $A = x_1, x_2, x_3$

or

(14) $\alpha_1 \; A \beta_1 \rightarrow \alpha_1 \; x_1 \beta_1$
$\alpha_2 \; A \beta_2 \rightarrow \alpha_2 \; x_2 \beta_2$
$\alpha_3 \; A \beta_3 \rightarrow \alpha_3 \; x_3 \beta_3$

Some information with respect to the context is given anyhow in the categorial notation and therefore (14) adds only a context "to the other side". It is clear, however, that (14) can be replaced by CF-rules. It would nevertheless be more convenient to drop (13) and have instead a rule that assigns a single category to A and then permutation rules that account for the various placements. Of course, once again, we would need some additional restrictions to be imposed on permutations.

A more serious problem arises if the categories are embedded in each other in some way and it is impossible to make use of the same "category label". One such possibility is

(15) $\ldots (X_{n-2} (X_{n-1} (X_n)))$,

the other

(16) $X_i \; A \; X_j$

where $1 \leq i, j \leq n$

and where A is a particular morpheme here A occurs in various contents .

It is easy to construct logical (semantic) examples for (15). Thus we may think of time and place as something that can be approximated with more or less accuracy and, furthermore, we may begin this approximation from nowhere, i.e., there is no limit where to begin it. This can also be realized in language. We may think of constructions like

> some time ago during a hot summer when I was in Mamaya I found myself one day, maybe it was Sunday, ... etc.

(15) can be made still more complicated by allowing for

(17) $$\ldots (x_{n-2} (x_{n-1} (x_n)))^r$$

where r is an integer. An illustrative example of (17) would be

> on 2nd of May, 1957, on the day of the great disaster and in the month when trees begin to get their leaves and, last but not least, the year of her first revelation, etc.

Here we have combined discontinuity and (15), i.e., potentially infinite approximation.

16 is the case when, in a given grammar, some morpheme A has infinite numbers of functions. This, as it has been remarked by both Bar-Hillel and Lambek[5], may happen in categorial grammar in case of such morphemes as "and", "only". Since this case is linguistically unmotivated one should try, maybe by an appropriate system of categories, to avoid it. In most cases some of the category assignment will be counterintuitive but by a later adjustment this can be remedied.

If we do not allow different functions then (15) is the case already tackled. The placement of certain morphems like adverbs may come under this heading.

While, then, (15) can be handled in some way within categorial grammar, for (17) we certainly would prefer ϑ to havetransformational rules. As transformational rules can be added to a system of dependency grammar, it must also be possible to use categorial grammar as the base. The base of generative grammar as propounded in Chomsky's Aspects of the Theory of Syntax consists of a CF-grammar. The context-sensitive rules being transferred to the lexicon, one has in the grammar another component containing transformational rules only. It is clear that most of the transformational rules can be rewritten as context-sensitive rules each transformational rule as an ordered set of CS-rules . In principle CF- and CS-rules can however be transferred to the lexicon. On the other hand, it seems plausible to assume that the base can be replaced by any grammar being weakly equivalent to CF-PS. If so, then we could speak about a dependency transformational grammar, categorial transformational grammar, etc.

In the case of (17) we might devise some set of CS-rules that would generate such structures. There are, however, structures that cannot be generated by CS-rules, e.g., the elliptic structures. Elliptical transformations deletion transformations cannot be reformulated as a set of CS-rules. Such transformations cannot be transferred to the lexicon; thus, they cannot be incorporated into categorial grammar. But it would be worth while to see what a categorial transformational grammar looks like even if it is clear that one comes very far by using phrase structure rules only.

REFERENCES

[1]This paper is part of a common project of the KVAL-group (Stockholm) and the Mathematical Linguistics Group at our Center.

[2]H. Karlgren - B. Svensson; An analysis of Swedish adjectives, KVAL PM 389 (1968).

[3]The examples are taken from Björn Hammarberg's paper in FSSB III.

[4]H. Karlgren, Syntactic calculus, KVAL PM 237 and 277 1967.

[5]See the references in F. Kiefer, A bibliography of categorial grammar, KVAL PM (forthcouring).

ON THE STRUCTURAL LINGUISTIC ANALYSIS OF POETIC WORKS OF ART

S. J. Petőfi

Contents:

1. Introduction

One of the most striking features of the present stage of the development of the sciences is the emergence of frames suitable to unify the various branches of investigation of a particular science -- depending on its level of development -- into an integrated whole. Partly the precondition, and partly the consequence of these frames is the fact that the particular branches have come closer together in both their structures and methods.

The process may be best observed in all its complexity in the development of mathematics. From statements of a definite content, in the course of the development of axiomatization, the axioms have become statement forms, and have thus become suitable for the elaboration of concepts which are well-defined but open as to their content. The set theory recognizing that which is common to the different concepts, has worked out for them operations easy to handle, thereby making possible the identical foundation of the various branches of mathematics. The structures interpreted with the help of arbitrary axioms and operations on any sets whatever have then led to the definition of abstract structures such as are open to different interpretations. As a result of all these the idea of building up the whole of mathematics structurally could arise, and in fact French mathematicians set about realizing this in the thirties of our century.

Although the development of mathematics is in itself of interest, what renders it even more so is the fact that the penetration of its methods into the other sciences has considerably contributed to the restructuring of several branches of science. To these belongs the science of language. It must be emphasized, however, that the transformation of linguistics -- the possibility of its establishment as an exact discipline -- has been made possible above all by the body of knowledge amassed by several hundred years of thinking about language and investigating languages.

The most characteristic criteria of the present development of linguistics is the endeavour to become exact and to co-ordinate the different aspects in linguistic research.

It was L. Hjelmslev who first tried to outline an exact linguistic theory on the basis of the doctrines of Saussure, and it was Noam Chomsky who in fact established the first exact theory of total linguistic description, the generative linguistic theory.

The generative theory of language, still being worked upon in its details, provides an all-embracing approach to all aspects /syntactic, semantic, phonological/ of the investigation of language. Syntax, semantics and phonology have been able to become components of a theory of language by becoming amenable to such treatment through the identical interpretation of their elements as being sets of certain basic elements through the establishment of generative and interpretative operations with these elements on the basis of a unified point of view.

This theory models that knowledge /competence/ of the members of a linguistic community which concerns the structures of sentences of their mother tongue and not their ability to construct actual sentences /performance/.[1]

This kind of transformation of linguistic theory lays also the foundations of the realization of the structural analysis of linguistic works of art to be carried out in a more exact manner than has been hitherto possible. /Here it must be stressed again that this new development has been largely assisted by previously amassed knowledge.[2]/

To our mind, structural analysis can become exact in this field only then if it strives after modelling the knowledge concerning the structure of the linguistic works of art, the process of exploring and describing the structures of works of art.

Besides promoting a fuller understanding of linguistic works of art /including a better understanding of the processes of creation and reception/ such an analysis is significant from other viewpoints as well. From these, we want to point out only one namely that the structural analysis of the linguistic works of art with the help of the latest achievements of linguistics is the primary intermediator of the structural method as a means for analyzing different and intricately structured systems composed of non-linguistic sings /for example works of art of a non-linguistic character/.

In this paper we want to deal with the problems of the structural linguistic analysis of poetic works of art. Before that, however, we must deal with a more general question.

2. About a structural theory of the analysis of linguistic works of art

In our opinion, the practice of structural analysis of linguistic works of art has reached such a phase that one can ask questions about its theoretical foundations.

In the following we try to outline a structural theory of the analysis of linguistic works of art.

Such a theory — in our judgement — must contain[3]:

1/ the definition of the linguistic work of art;

2/ the definition of the possible structure types of the linguistic work of art;

3/ the definition of the structural description of the structure types

4/ the description of the models making possible the structural descriptions;

5/ the definition of such methods by which, on the basis of the defined models, the structural descriptions may be carried out.

1. To the definition of the 'linguistic work of art'

The term linguistic work of art will be temporarily used by us as a non-defined notion. In order to explain the other theory-components, however, we have to summarize its main characteristics.

Every linguistic work of art can be regarded as a single sign with a particular structure.

The sign, according to its most general definition, is a solidary unity of two sign components, the signans and the signatum /mutually depending on each other/. The character of the signs is defined on one hand by their signans and signatum, on the other hand by the form of their unity and interdependence. A system of signs of an identical character is called 'semiotic system'. The totality of the linguistic works of art so can be regarded as a particular semiotic system.

Regarding the structure of its sign components, a linguistic work of art is such a sign, the signans of wich is also a sign itself, that is a 'linguistic communication'. This linguistic communication has a 'direct meaning' but, within the linguistic work of art, this direct meaning structured in a given way is at the same time the form of expression of an indirect secondary meaning, i.e. of the artistic message.

Regarding their signans, linguistic works of art belong to the so-called secondary modelling systems, which are — according to Lotman's thesis — "such structures that are based on a natural language but in addition to this the system possesses also secondary, supplementary structures which can be of ideological, ethical, poetical or some other kind of character."[4]

Here we cannot speak of the solidary unity of the signans and the signatum. 'Literary work of art'-signs cannot be registered into a dictionary. The meaning or, to express it more exactly, the possible meanings cannot be rendered independent of the recipients. The exploring of this depends on the personality and knowledge of reality of the recipient and also on the individual and social conditions of the reception. This is the consequence of the fact that linguistic works of art are open from certain aspects.

The structure of the linguistic work of art as a sign may be illustrated by Figure 1.

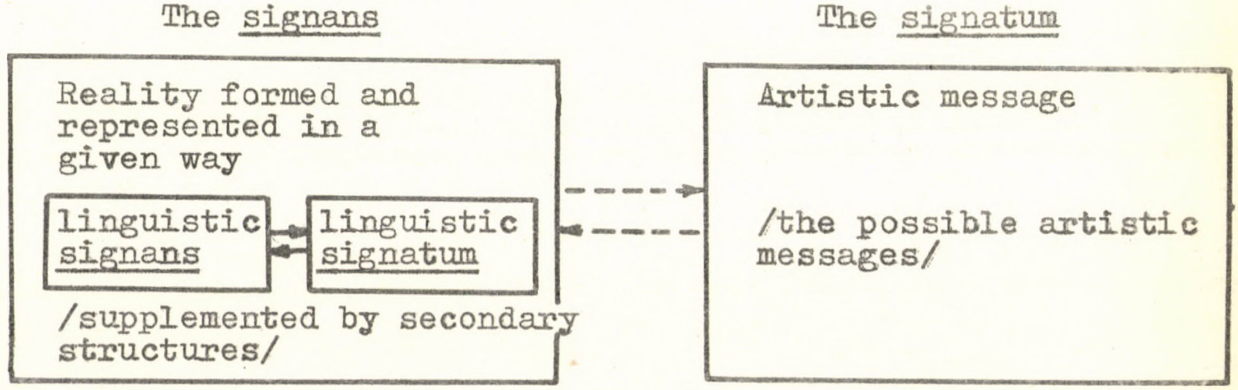

Figure 1.

2. <u>To the definition of the 'possible structure types'</u>

If we consider the linguistic work of art as a sign from the different aspects of its signans, we shall get the possible structure types as a result. /We should like to emphasize that here we think of the work of art <u>taken in itself</u>./

As to the character of the linguistic form of the signans /the linguistic signans/ we can make a difference between <u>poetry</u> /poetic works of art/ and <u>non-poetry</u>.

Here we use the term'<u>poetic work of art</u>' as a synomym to the <u>linguistic work of art in verse</u>. The verse--form, the verse -- accepting provisionally De Groot's definition[5] -- is "a text consisting of continuously corresponding units /called "lines"/ which are sequences of words and at the same time variations of a more or less constant auditory theme".

Non-poetry, or in this relation **the artistic prose** — according to Lotman's statement — though it is seemingly closer to the everyday usage of the language, is aesthetically more complicated than poetry, its simplicity is only a secondary phenomenon. In order that simplicity should really be simplicity and not primitiveness, it is necessary that it should be simplification, that is, the artist should not use certain means of art, and do it intentionally, and the recipient should be able to project the simple on such a level where these means have been used. Hence, artistic prose is: text + the lack of the strictly regulated artistic devices of poetry.[6]

Structure types resulting from the projection of the linguistic and the secondary structures one upon the other, are illustrated by Figure 2.

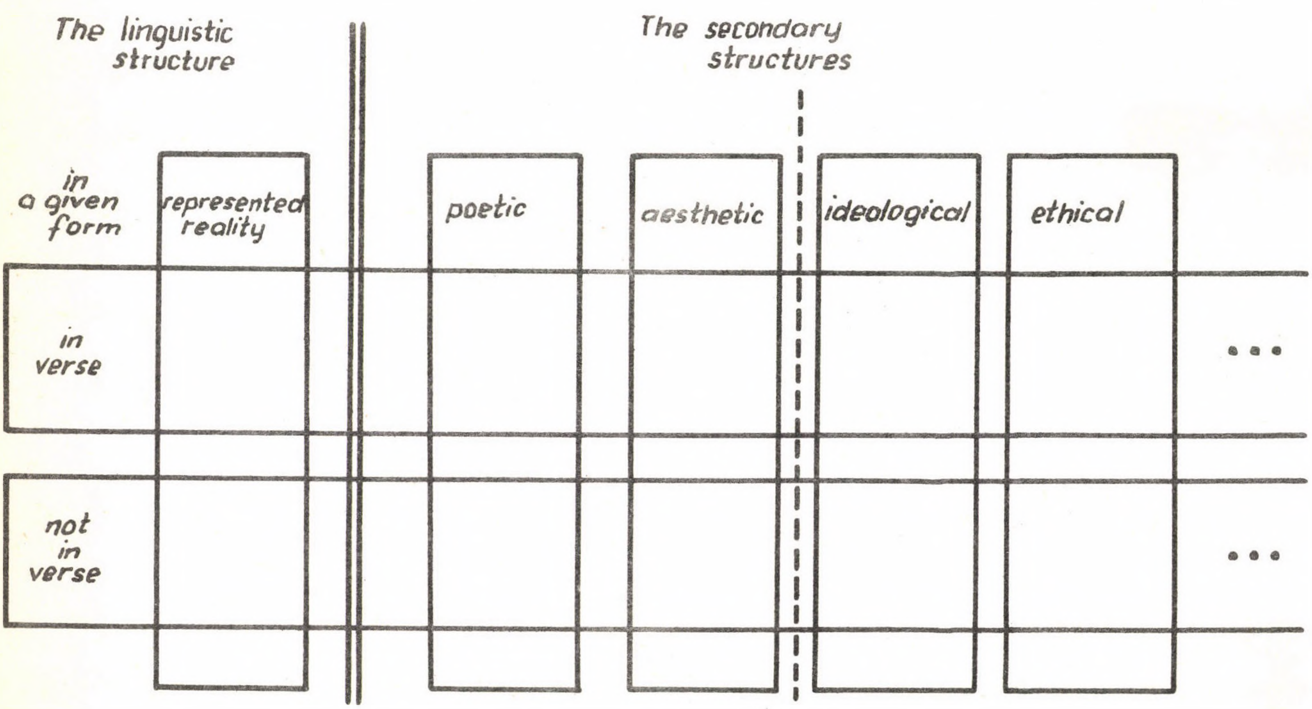

Figure 2.
The different structures of the 'signans' of linguistic works of art

All of the common segments of this cross classification can serve as an object to a structural description. /In every case of structural analysis it must be cleared up that what kind of a structure is to be analyzed./

The fundamental and primary task is evidently the exploring and describing of the linguistic structure of the signans because the analysis and description of the secondary structures is based upon the former.

Reality formed and represented in a given way in a work of art -- the linguistic structure of the signans -- is, in general, in the same way given to every recipient belonging to the same linguistic culture. The exploring and describing of it can be solved in such a way that we remain within the sphere of the given work of art and use only one external system, the grammatical system /syntax, semantics and phonology/ of the given language.

The exploring of the different secondary structures can be carried out only then if we have a knowledge of the linguistic structure and certain other structures outside the sphere of the given work and differring from the grammatical system. After describing the linguistic structure in an exact way it is presumable that an exact theory of the description of the poetical and aesthetical structures can also be worked out.

x x x

Since it is our intention to deal exclusively with the analysis of the linguistic structure of the signans of poetic works of art /further simply 'linguistic structure'/ we interpret all other components of the structural theory of the analysis of literary works of art only in connection with this kind of structure.

We want to outline such a theory which "knows" all that what, insofar, all the traditional and present structural methods have known about in the following interpreted structure of a linguistic work of art.

However, we wish to explore this knowledge with a general algoritmic method unifying all wiewpoints instead of approaching every single structure in a partial and intuitive way.

The analysis of a poetic work of art is necessarily longer than the original work itself. The set of the possible signans structures, being on the level of a work of art cannot be generated, the internal structure of every single work must be explored and described individually. Consequently the size of the analysis is always defined by the analyzed work. The method of the structural description must make it possible that the description should be able to take into consideration all relevant characteristics of the signans structure, and only those and nothing else.

3. The definition of the 'structural desciption' of the linguistic structure.

1. The interpretation of the term'structure'.

We can differentiate the following sign-somponents of the linguistic structure of the poetic work of art /on the basis of the multistructured character of the linguistic sign/: semantic, syntactic, rhythmic and phonetic. Let us call the first two linguistic and the second two musical sign-components. /We speak about sign-components because the signans of the linguistic structure is formed of linguistic signs./

In the linguistic sign-component we examine the sequences of meaningful elements while in the musical sign--component only the sound-texture of them. /Since the musical sign-component contains elements which can be interpreted from the phonological-phonetic aspect of the language, the analysis of this component is essentially also of linguistic character./

The term structure, in its general meaning according to which "against a simple combination of elements a whole of interdependent phenomena is marked by it, such a whole, in which every element is depending on the others and has its own particular existence only and exclusively in its relation to the others"[7], will be used accordingly in the following sense.

The whole of interdependent phenomena has a double meaning. On one hand it relates to the linguistic and musical sign-components, and does it separately, on the other hand to the work itself, as to the whole existing in the unity of these two sign-components.

The interpretation of element, within every single sign-component and within the whole of the work of art, is the following:

We regard the linguistic communication unit as the element of the linguistic sign-component, while as the element of the musical sign-component -- being on the same level as the linguistic communication unit -- we regard the musical communication unit.

The element of the work of art as a whole existing as the unity of two sign-components, is a correlative pair built up by the communication unit of one of the sign--components and by the respective segment of the other sign--component. In this correlation now, the linguistic, now the musical communication unit is the dominant.

The text formed in the given way is regarded as the one establishing and exclusively expressing the relation between the elements within the single sign-components and also within the whole of the work of art. /The explication of these notions comes later./

2. The definition of the 'structural description'.

The description of the signans structure of a poetic work of art as a unity of the linguistic and musical sign--components means the discovering and describing its hierarchical and linear patterning and the individual semantic-network created by these.

This is to be understood as follows:

1. The hierarchical patterning includes two such layers which can be separated in a rather definite way in both of the sign-components, the layer of the composition units and that of the communication units.

The segment consisting of interdependent communication /or composition/ units of the linguistic work of art is called a composition unit.

Within the layer of the composition units existing in the single sign-components we have to point out how the work, the composition unit on the highest level, breaks up into smaller composition units, and these into more smaller ones, and so on, until the whole process reaches the level of the communication units. While in the layer of the communication units we have to explore and illustrate the way they are composed in and depict the character of the communication unit constituents which build them up. We cannot speak about constant levels in any of the layers.

In the linguistic sign-component existing in the layer of the composition units, the structure of the represented reality is the dominant, while in the layer of the communication units, the grammatical structure.

2. The description of the linear patterning means the discovery and listing of paralellisms taken in a most general sense in both of the sign-components, the irregular or periodic returns of units belonging to different levels, and also that of the character of the returns. /The character of the return is either the identity or analogy./

"An objective, careful, exhaustive and full description of the selection, distribution and interconnection of the morphological classes and syntactic constructions in a given poem -- states R. Jakobson -- surprises the investigator himself by unexpected, conspicuous symmetries, balanced structures, effective accumulation of equivalent forms and striking contrasts; and last but not least by the strongly limited set of the morphological and syntactic elements made use of in the poem; on the other hand even this selection makes it possible to follow the masterful interaction of the actually used components."[8]

Furthermore, it could be useful to determine the different statistical characteristics of certain units of the single sign-components /to find out the so-called type-token relations or some other types of information theory characteristics/. The relevancy of these characteristics depends on the length of the text, too.

3. When exploring and describing the linguistic structure of the poetic work of art we have to start out of the work as a whole, and have to point out that what kind of a particular linguistic, linguistic-musical semantic network results from the linearly and hierarchiacally structured correlation of the two sign-components, the formation of the dominant character of the two sign-components, and that of the two different ways of being structured. /This semantic network is the medium of the different secondary structures./

4. The scheme of a model making possible the structural description of the linguistic structure

As we have several times emphasized we are dealing only with the possibilities and methods of the description of the linguistic structure. We have already mentioned also that fact that the set of the signans structures of the poetic works of art -- not like the set of the possible sentence structures -- cannot be generated.

For the comprehension of this it is enough to consider the followings:

In the linguistic sign-component

a/ the way of decomposing poetic works of art into different composition units cannot be defined by a limited number of rules,

b/ the primary elements -- the communication units -- are not given like in other sign systems, they are single and unique /created by the artist in each case separetely/, they cannot be classified within a limited number of classes and, consequently, categories of a limited number cannot be substituted for them.

In the musical sign-component -- not regarding dertain fixed forms -- the situation is just the same.

Taking into consideration all this and the construction of the linguistic work of art as a sign, it is evident that the linguistic structure of the poetic works of art can be explored and described only with the help of a complex model. Such a complex model has to include the following components.

1. Analyzing components

We need two types of analyzing components in compliance with the twofold ways of being structured. These can be so regarded as the syntactic component of the model. The first analyzing component constitutes the basis for the model. Its primary task is to analyze the hierarchical structure of the single sign-components.

The task of the second analyzing component is to analyze such relations of different units which establish a paralellism.

2. Interpreting component

It can be so regarded as the semantic component of the model which on the basis of the results of the preceding component carries out the semantic interpretation of the work from a linguistic-musical aspect in first approach.

3. Structure describing component

The function of this component is, starting out of the given poetic work as a whole, to produce the description of its structure on the basis of the units and connections

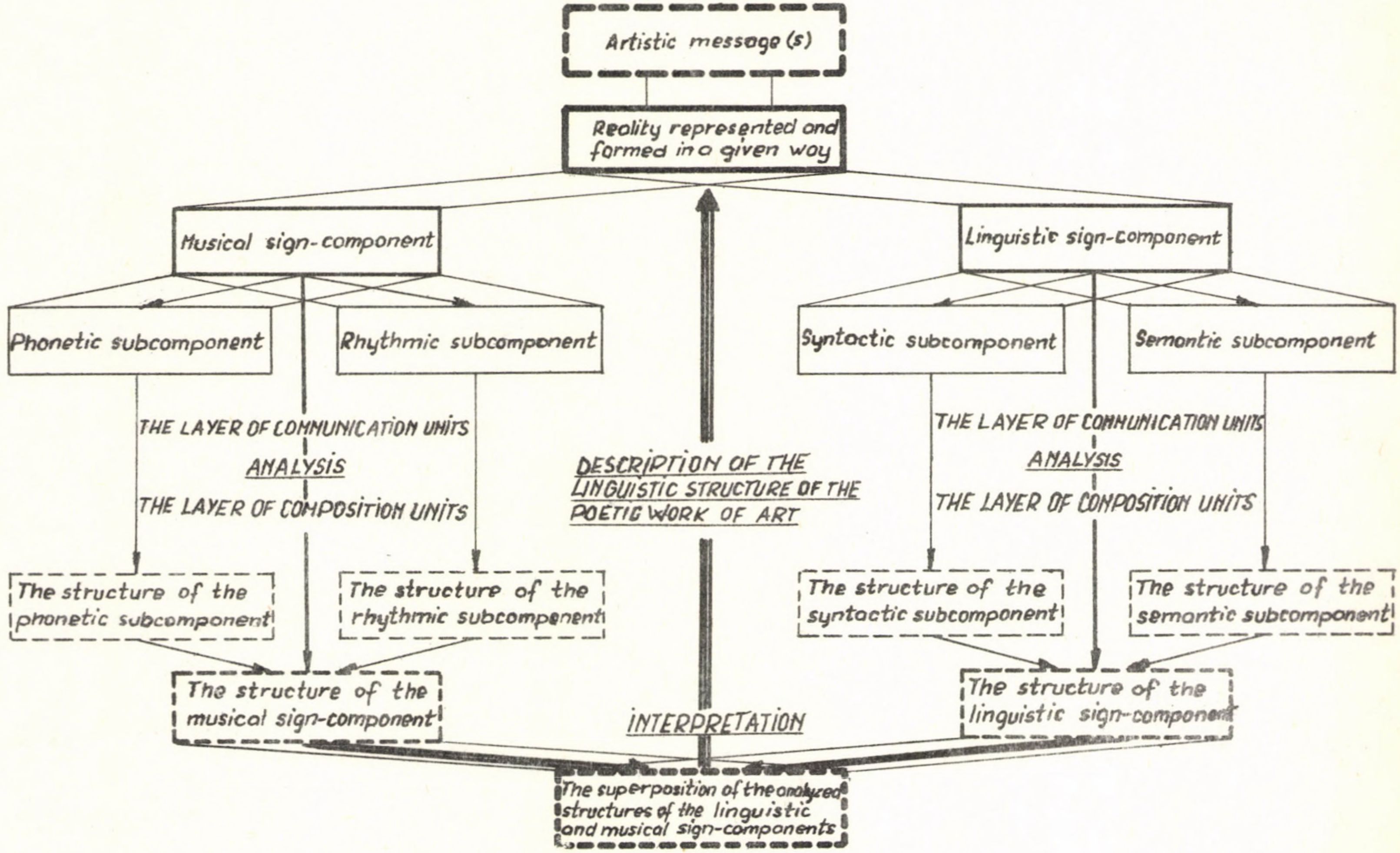

Figure 3.

AN OUTLINE OF THE DESCRIPTION OF THE LINGUISTIC STRUCTURE OF POETIC WORKS OF ART discovered by the analyzing and interpreting model components. This can be regarded as the total semantic interpretation of the linguistic structure of the work to be rendered by the given model. /See Figure 3/

5. A possible method for exploring and describing the linguistic structure

While the above outlined model so can be regarded as a definite frame for the discovering and describing the linguistic structure, the methods of exploring and describing can be of various kinds.

Since the most comprehensive theory of the description of the linguistic structures is the generative linguistic theory of Chomsky, we consider it advisable to use this as a starting point, supplemented by a proper text theory. Such a method can be outlined in the following way.

1. Analysis

The first task is to decompose the work into communication units. We obtain the linguistic communication units of the work in such a way that we segment the text on the basis of the definition of the linguistic communication unit /which must be defined by the theory/. In the musical sign-component we decompose the text into elementary units /into linguistic-musical syllables/ Then taking into consideration the stress relations, the word, construction and communication unit boundaries of the linguistic units and also the breaking up of the work into lines, we establish the musical communication units.

After defining the communication units we analyze the relations of the communication units following each other, one after the other. Then, we analyze the sequence of the such defined composition units until we reach the final composition unit, the work itself.

In the course of establishing the units we add to the constituents of the communication units and to the communication and composition units in both sign-components categories relating to the hierarchical and linear patterning, so-called compositional markers.

The markers of the hierarchical patterning are partly of paradigmatic, and partly of syntagmatic character. The former characterize the single elements within the class of the elements of identical character, while the latter express their function what they perform in the next higher unit. One part of them is general and characteristic to the language itself, another part of them has an individual character and relates only to the given work.

The work is - by definition - the composition unit on the highest level. There are four basic types of the ways in which the communication units as well as the composition units can be organized into higher units until we reach the work itself:

	1.	2.	3.	4.
explicit syntactic linking	+	+	-	-
explicit semantic linking	+	-	+	-

In the case of the last type, i.e. where there is neither syntactic nor semantic linking, the way of being organized into the work cannot be established by the help of linguistic analysis.

The indices of the linear patterning concerning every unit mark the sequence of the next lower level units, the constituents of the given unit which, as it is well-known, plays an important role in the preparation of the expectation for the appearance of elements of different character, and in the satisfying and breaking this expectation, respectively.

After this, in both sign-components we add the indices of such units to every composition, communication or smaller units, which are identical in one or more elements of the markers and indices concerning the given sign-component and on this basis we examine the relations of the units having identical elements. /When analyzing the relations -- with regard to the limited capacity of memory -- we have to take into account the distance between the units related to each other./

2. Interpretation

After the analysis we match the analyzed structures of the linguistic and the musical sign-component and establish the correlations forming the units of the work and also the relations of dominance within the correlations of the linguistic and musical units.

Then we unite the compositional markers belonging to the two sign-components and also the indices of the linear patterning based on the dominant communication and composition units. Following this procedure we explore the semantic bearing of the different units and the dominance relations of the two types of patterning as they form within the work.

3. Description of the linguistic structure

The description of the linguistic structure is the description of the formerly explored semantic network starting now from the work as a whole, thus, from the direction just opposite to that of the analysis. In the case of this description we project the semantic structure concentrated in the course of the analysis and interpretation on the peak of the hierarchy of the units upon every element.

x x x

6. Examples to illustrate the notions used in the definition of the structural description

To define in an exact way the terms 'communication unit' and 'composition unit' is the task of the theory worked out for the analysis of the linguistic and the musical sign-component, respectively.

Our only aim with the following examples is to illustrate the contents of these notions and to demonstrate the character of the hierarchical and linear patterning.

Remarks:

a/ When selecting the verses we strived to choose such ones by which it would be possible to demonstrate not only the process of breaking up into communication units but also a great number of the correlation possibilities between the linguistic and musical communication units.

b/ The poem Window in the Night by Sándor Weöres is an example to such a verse-type, to the decomposition of which the author does not give any help by breaking up the text into sentences.

In such cases every possible segmentation must be analyzed. If there are more than one possibilities, the phenomenon is called compositional homonymy. Here we demonstrate only one of the possible segmentations.

c/ Those informations lay the foundation of the musical decomposition which are contained by the written text suggesting the sound shape. Consequently, we have to take into consideration not only the logical stress of the single words but within the corresponding model components the relative stress conditions within the constructions and between the connected constructions. The lines of the verse influence the decomposition by their stress conditions and sound texture.

d/ At the demonstrated poems lg denotes the linguistic, m the musical communication units, while Lg and M the linguistic and the musical composition units, respectively.

/1/

	Őszi dal		Autumnal song
M M_1 m_1	Őszi ködben	lg_1 Lg	In the fog of autumn
	zúgó ötven		booming fifty
	nyárfa,		poplars,
m_2	ötven dal van		fifty songs are
	<u>törzsetekbe</u>		closed
	zárva.		<u>in your trunks.</u>
M_2 m_3	Őszi csöndben	lg_2	In the silence of autumn
	nyíló ötven		opening fifty
	láda,		cases,
m_4	ötven szív van		fifty hearts are
	<u>deszkátokba</u>		closed
	zárva.		<u>among your planks.</u>

/Sándor Weöres/

About the linguistic /Lg/ and musical /M/ the hierarchical /H/ and the linear /L/ patterning of the demonstrated poems and about the correlation of these two sign-components and their superposed structure /SS/ we want to underline the followings:

1. Autumnal song

LgH: The work breaks up into communication units in a direct way.

LgH: lg_1 and lg_2 -- also possessing several common lexical elements -- have an entirely identical grammatical structure.

MH: Every two communication units make up one composition unit /M_1, M_2/

ML: Both composition units have an identical structure which, by the help of a syllable structure, can be put down as follows /blanks corresponding to word boundaries/:

2 2 2 2 2

2 1 1 4 2

SS: The structures of the linguistic and musical sign-components are concurrent. In the poplar, which is booming, songs are closed. In the opening case hearts. And the four syllable words in m_2 and m_4 making impossible the segmentation according to the pattern of m_1 m_3 -- in your trunks, among your planks -- mark the concrete place of being closed. /It is rather interesting to observe, how this fact is emphasized by the written form of the poem./

This concurrent structure links the two units.

2. World of the reeds

LgH: The composition units following the highest composition unit, that is the work as a whole: marking of a part of the world of the reeds /Lg_1/ -- the water-chicken herd /Lg_2/ -- the double image reflected by the water mirror

	Nádi világ				World of the reeds
M m_1	Éren-nádon sikló kúszik	lg_1	Lg_1	Lg	Grass-snake worms its way in the brook and among the reeds
	kicsi patak-ágyon vizicsibe úszik.	lg_2			and on the rivulet-bed water--chicken is swimming.
m_2	Hajló nád közt kotlós zizzen,	lg_3	Lg_2		Amongst rippling reeds brood-hen gives a sudden rustling sound,
	vizicsibe-népét tereli a vízben.	lg_4			driving her water-chicken folk in the water.
m_3	Ér tükrében látszik az ég is	lg_5	Lg_3		In the mirror of the brook also the sky is to be seen,
	fejetetején a vizicsibe-nép is.	lg_6			upside down also the water--chicken folk.
m_4	Siklók, békák, pókok látják		lg_7		Grass-snakes, frogs, spiders see
	vizicsibe-pásztor vizicsibe-nyáját.				the water-chicken herd of the water--chicken shepherd.

/Sándor Weöres/

/Lg_3/ -- the return of the marked part of the world of the reeds /lg_7/.

LgL: The cohesive role of the lexical elements what they play in connection with the composition units, is most conspicuous: besides the element of the <u>water-chicken</u>, which can be found in every composition unit, the others commence with one of the elements of the first composition unit: Amongst rippling <u>reeds</u> ... In the mirror of the <u>brook Grass-snakes</u>, frogs ...

MH: There are no interim composition units between the work and the communication units.

ML: The communication units have almost identical structure only m_3 is a protuberant variant.

SS: The structures of the linguistic and the musical sign-components are concurrent. The sound texture and the rhythmics of the variants of the <u>water-chicken</u> theme, and those of the second members of the musical communication units are organic parts of the contents of the work. The "protuberant variant" m_3 is in a close correlation with the "double image reflected by the mirror of the brook".

Against the predicates expressing actions of the other communications -- <u>worms</u>, <u>swims</u>, <u>gives a sudden rustling sound</u>, <u>see</u> -- here the sky and the water-chicken folk <u>are to be seen</u>, but not so as everything else mentioned in the poem, but <u>upside down</u>.

3. <u>Window in the night</u>

LgH: The work disintegrates into two composition units in a direct way /Lg_1, Lg_2/: the total communication units relating to the moon /lg_1-lg_5/ -- their transformed and broken, fregmentary returns lg_6 /lg_3/, lg_7 /lg_6/, lg_8 /lg_2/, lg_9 /lg_4/. Lg_1 disintegrates again into two smaller composition units: one /Lg_1^1/ is characterized by third person communication, the other /Lg_2^1/ by the second person addressing form.

	Ablak az éjbe		Window in the night
M M_1 m_1	Szűz arc - égi vánkosán -	lg_1 Lg_1^1 Lg_1 Lg	Virgin face - on its haevenly pillow -
	át a vércsék városán		through the town of windhovers
	lehajolt.		bowed down.
m_2	Szárnyak és csőrök felett	lg_2	Over wings and beaks
	gömbölyű tér megreped.		round space bursts.
	Penge hold!	lg_3 Lg_2^1	Blade moon!
M_2 m_3	várás ablaknégyszöge,		window square of waiting
	bámuló illat, zene		staring odour
	libegő		flickering
m_4	kerete négy széle fog,		frame four edges of music holds,
	szélből épült bástyafok,		parapet built of wind
	égi kő.		stone of heaven.
M_3 m_5	De ha te hámba fogod,	lg_4	But if you harness,
	nyergeled az ablakot.		saddle the window.
	Paripád,	lg_5	Your steed,
m_6	vihar villám szorosan		storm lightning canyon through
	-- fénylő szűz arc -- eloson		-- bright virgin face -- steals away
	ködön át.		through the mist.
M_4 m_7	Ablaknégyszög, aki vá...,	lg_6 Lg_2	Window square who wai..., /from waits/
	illat, zene, aki bá...,		odour, music who sta..., /from stars/
	kerete...		its frame...
m_8	Illa-bá és villa-bé	lg_7	.... and ....+
	Az éj meghasad belé	lg_8	the heaven splits...
	de ha te...	lg_9	But if you...
	/Sándor Weöres/		

+This line presents an untranslatable word-play.

LgL: The communication units of the composition unit Lg_1 are held together by a thesaurus-like connection manifested in the lexical elements:

lg_1 <u>Virgin face</u>
through the town of windhovers
lg_2 Over wings and beaks
round space bursts
lg_3 <u>Blade moon!</u>
window square of waiting
lg_4 <u>you</u> .. saddle the window
lg_5 <u>Your</u> steed, ..

-- <u>bright virgin face</u> --

In connection with the semantic structure of these communication units we have to emphasize that the peculiarity of certain lexical elements' getting side by side cannot be explained by linguistic means and methods.

/From the contexts we can only render it probable <u>in an associative way</u> that certain lexical elements impressing us as particular, what kind of elements impressing us as natural do they stand for. The lg_3 gives us a detailed description of a sequence of associations: <u>window square of waiting</u> -- <u>flickering frame four edges of music</u> -- <u>parapet built of wind</u> -- <u>stone of heaven</u>./

The succession of the communication units of the composition unit lg_2 is partly defined by the first composition unit.

MH: The rhyme structure builds up single composition units of every two communication units.

ML: The musical communication units are the variants of the same primary structure

SS: The linguistic and musical communication units penetrate into each other in such a way, that in the first composition unit of the work which is the first linguistic

	Négysoros		Four-line poem
M m_1	Alvó szegek a jéghideg homokban.	$lg_1 Lg''Lg'Lg$	Sleeping nails in the ice-cold sands.
	Plakátmagányban ázzó éjjelek.	lg_2	Soaking nights in poster-solitude.
m_2	Égve hagytad a folyosón a villanyt.	lg_3	You left the lights on in the corridor.
	Ma ontják véremet.	lg_4	Today will be my blood shed.

/János Pilinszky/

composition unit /Lg_1/ the linguistic, while in the second, the musical sign structure is the dominating.

The two composition units are linked together by the already mentioned "fregmentary return". /Here have we to mention also the significant role of the titles of the poems./

4. Four-line poem

LgH: The part of the text composed by the first three communication units of the work /Lg'/ is such a composition unit which disintegrates into two further parts /Lg" and lg_3/.

Among the first three communication units a certain indirect thesaurus-like relation can be manifested but this concerns both linguistically and from the viewpoint of the represented reality only single lexical units and not the whole of the communication units: Sleeping nails ... soaking nights in solitude, you left the lights on ...

The units Lg" and lg_3 are defined by the character of the predicate /in lg_3 the second person addressing form/.

The fourth communication unit does not join with the first three, not even in a thesaurus-like manner.

LgL: The linear structure is characterized exlusively by the thesaurus-like connection.

MH: The work breaks up directly into communication units

ML: m_2 is the variant of m_1, its second member is broken, incomplete. The number of their syllables: 11+10 -- 11+6.

SS: The lack of connection between lg_4 and the preceding three communication units mentioned above, emphasized also by the second, broken member of m_2, indicates, that the interconnection, the basis of composition, is to be searched in something else.

The first three linguistic communication units are the symbolic indication of such contents /feeling, mood/ which lg_4 joins already in an organic way.

Here we should like to emphasize once more that our only intention was to illustrate the notions used in the 'definition of the structural description', and by no means can these remarks be considered as the description of the whole analyzing, interpreting and structure--describing process.

7. Conclusion

In our paper we have tried to outline a model capable for exploring and describing the linguistic structure of poetic works of art. To our mind, the importance of outlining a complete model lies, among others, in that fact that those in an autonomous way developed or developing branches of science which approach a poetic work of art only from one aspect /only the music of its language, its rhythmic, or its linguistic form etc./ acquire quite another kind of perspective in the mark of the endeavour to explore the connections in their totality. This perspective making possible the convergent development is that, which could lay a foundation for a complete model to be worked out. However, this can be realized only gradually.

The first and most important step is to work out the methods of the analysis. This means that the following tasks have to be accomplished: to define the linguistic communication unit, to work out such a syntax, semantics and phonology which consider also the particular characteristics of the communication units, to establish a text theory based on the communication units /compositional syntax and semantics/, to define the musical communication units, to work out a method capable for the analysis of the rhythmic and sound--texture of the musical sommunication units, to create a language-musical text theory based on the exact description of the given language /composition language-musical syntax/.

When solving these problems we have to consider the primary condition of the operation of the interpreting component according to which the analyzed structure of the linguistic and musical components should be capable of being superposed. This means that -- within the limits of possibility -- the structure of all ingrediants of the analyzing component must be identical. The analyzing and interpreting methods must be worked out in a constant correlation between the two.

If considering the description itself, the analyzing and interpreting operation line does not belong to the actual description of the linguistic structure, /From this point of view this can be so regarded as a work behind the scenes./ However, its methods and results are from other viewpoints significant as well, not only from that of making possible the description of the structure. It can be expected that they will render a lot of information concerning the usage of the language as well as the process of creation/ reception.

In the description the phenomena which cannot be interpreted within the frame of the linguistic stracture have the same importance as those which can be interpreted.

While the interpretation of a structure which can be described also in itself is only enriched if we put it in among broader relations of the same level, this procedure is absolutely necessary if we want to interpret such linguistic structures which can only be partly or not at all described. One of the results of the above described model will that be that it will show the boundaries of the territory of those correlations which can be explored by the analysis of the linguistic structure.

The description of the linguistic structure will make it possible to establish such a structure typology in the future which will serve as a solid foundation for the exact way of realization of the poetic and aesthetic interpretation.

REFERENCES

1. See N. Chomsky, Aspects of the Theory of Syntax, Cambridge, Mass. 1965.

2. Here we have primarily linguistic approaches, papers of R. Jakobson, S. R. Levin, Ju. Lotman, M. Riffaterre etc.

3. In connection whit this see:
N. Chomsky, op. cit.
M. Bierwisch, Poetik und Linguistik, in: Mathematik und Dichtung, München, 1965. pp. 49--67.

4. See the paper of Ju. Lotman, in: Trudy po znakovym sistemam II. Tartu, 1965. pp 22--37.

5. A. W. de Groot, The Description of a Poem, in: Proceedings of the Ninth International Congress of Linguists, The Hague, 1964. pp 191--197.

6. Ju. Lotman, Lekcii po struktural'noj poetike. Trudy po znakovym sistemam I. Tartu, 1964.

7. V. Brøndal, Linguistique structurale, in: Acta Linguistica /Copenhaga/ 1/1939/ pp 2--10.

8. R. Jakobson, Poesie der Grammatik und Grammatik der Poesie, in: Mathematik und Dichtung, München, 1955. pp 21--32.

ON A SELF-INTERPRETING INTERPRETER

Gy. Révész

1. Introduction

The machine language of electronic digital computers are established in such a way as to make simple the logical design of computers and particularly that of their control units. There exist on the other hand a great deal of programming tools that are not built into the computers, the so-called software.

Professor L. Kalmár [1] and others /see e.g. [2] , [3] / have suggested that computers should not remain at this stage of evolution but they should have much better machine languages in the future. They have shown that it is possible to built machines that could be programmed in more efficient languages. Recently in the USSR a scientific computer has been designed whose machine language is ALGOL-60.[1]

At present, the use of highly developed algorithmic programming languages demands different translators /compilers/ or interpreters for any given pair of algorithmic and machine languages. Much work has been done to construct compilers that are machine-independent as far as possible, i.e. they get rid of the special features of the machine language used. This is, however, possible to some extent only, because the translation process depends perforce on both the source and object languages.

---

[1] The language of machine МИР was published in 1965 [13] , while about the more advanced УКРАИНА I was informed personally by L. Kalmár.

It is possible to establish a common machine-oriented language as an intermediate language. In this way the translation process will be divided into two steps. The first step is the translation from the source language into this common machineoriented language the second step being the translation from the latter into a concrete machine language. Here the first step is machine independent and only the second step must be different for each machine. The concept of ALMO represents an effort in this direction.[2]

On the other hand any change of the source language requires the alteration of the translator as well. The idea of mechanizing these changes have led to the constructions of Syntax Directed Compiler [4] and Compiler Compiler [5] .

One may ask whether one can go further in mechanizing the construction of translators. One may imagine a "compiler generator" which could accept the formal definitions of both the source and the object languages and would be able to produce the corresponding compiler. This solution seems to be quite simple if we ignore the great difficulties of defining independently and correctly the syntax and semantics of two different languages. Many difficult problems arise in this connection apart from the question of efficiency. In fact we need a third language for defining independently the semantics of the two languages etc.

The problem will be clearer if we take into consideration that according to some experts the compiler of a programming language could be considered as the best syntactic and semantic definition of that language while the definition of a machine language has to be given by the description of the machine that

---

[2] ALMO is the abbreviation of algorithmic machine oriented language [14] .

interprets this language [6] , [7] . The concepts of defining machine and defining compiler as proper tools of language definition are both correct and practical. But in connection with the "compiler generator" this would mean that we should have to make a compiler in order to enable the "compiler generator" to do the same which, of course, is nonsense. It is really not less difficult to define exactly an algorithmic language than to construct its compiler. In fact any form of an exact definition of a programming language must be equivalent to a compiler or interpreter of that language.

In the present paper we shall give the definition of an algorithmic language by means of its interpreter written in the same language. Thus we have only one formal language and our interpreter represents a hypothetical machine that is our defining machine. In fact we can construct such a machine on the basis of the description of the interpreter. This would be a kind of a formula-directed computer. On the other hand, our definition is completely machine-independent.

In other words we define our language with the aid of a special text of that language which represents the interpreter i.e. the program of interpretative execution of any text /algorithm/ of the language. This interpreter plays the same role as the universal algorithm in the theory of algorithms or as the universal Turing-machine /see e.g. [8] and [9] /. It is just one algorithm but it can simulate the execution of any other algorithm written in the same language.

Having this interpreter we can "teach" our language to any computer by translating just this single text into the machine language of the given computer. Thus the problem of translation from our language is reduced to the translation of one of its texts.

This text must have a meaning in order to be used for defining the language so we must define at least a given part of the language before being able to grasp the syntax and semantics involved in that special text.

Accordingly we have two levels of definition, the first of which we may call a priori the second part a posteriori. The a priori definition will be given in a similar fashion the ALGOL-60 is given in the ALGOL reports. [10] , [11] . The a posteriori definition, i.e. the interpreter, represents the way of reaction of our hypothetical machine to any text fed into it. Therefore we might consider it as if it could be experimentally checked. The two levels of definition do not contradict each other, they really supplement each other.

2. The syntax of the algorithmic language L_{si}

We have chosen for simplicity' sake an ALGOL-like but much simpler language. First of all we excluded the block-structure and the procedure concept of ALGOL-60. Our language is, with some exceptions, a part of the ALGOL-60. We give the syntax of L_{si} in Backus-notations as follows:

Basic symbols

<basic symbol> ::= <declarator> | <operator> |
<sequential operator> | <bracket> | <separator> |
<input-output operator>

<declarator> ::= <type> | switch | array

<type> ::= integer | real | Boolean | common

<operator> ::= <arithmetic operator> | <relational operator> |
<Boolean oparetor> | <function designator>

<arithmetic operator> ::= + | - | × | / | ↑

<relational operator> ::= < | > | = | ≠

<Boolean operator> ::= ¬ | ∧ | ∨

<function designator> ::= ent | stand

<sequential operator> ::= go to | if | then | else | for | do |
jump | exit | stop

<bracket> ::= (|) | [|] | begin | end

<separator> ::= , | : | ; | := | step | until | ⇐

<input-output operator> ::= read | print | input | output

The common type is allowed only in the copy statement. We shall discuss it later.

Operands

Beside the basic symbols we define three more basic categories which are called operands. These are:

<operand> ::= <logical value>|<number>|<identifier>
<truth value> ::= true | false
<number> ::= <integer number>|<fraction>

Here we must explain that this language constitutes an internal representation. We suppose that all the basic symbols are coded somehow, and also the operands will be coded during the reading of the program. Let us say that we have a special input program which reads the program to be interpreted from the punched tape and places it into the core storage of the computer. The punched tape contains the program in an external representation which is of no interest at this moment. But in the core storage we shall have it in the internal form and that is precisely the language we are dealing with. The details of the coding of the operands are, for the moment, also uninteresting. The only important thing is that the operands are all distinguishable from each other and from the basic symbols and recognizable at each place of their occurrencies. Later we shall give a detailed description of the internal coding of the language required by the interpreter.

Declarations

<declaration> ::= <type declaration> | <array declaration>|
<switch declaration>
<type declaration> ::= <type> <identifier list>
<identifier list> ::= <identifier> | <identifier list> ,
<identifier>
<array declaration> ::= <type> array <array list>
<array list> ::= <identifier> [<extension>]|<array list> ,

```
                  <identifier> [ <extension> ]
<extension> ::=<integer>|<extension> , <integer>
<switch declaration> ::= switch <identifier>
                 [<identifier list>]
```

As can be seen from the array declaration, the extension of an array is defined by integer numbers which represent the upper bounds of the corresponding subscripts. The lower bound of each subscript is assumed to be 1.

<u>Variables</u>

```
<variable> ::= <simple variable>|<subscripted variable>
<simple variable> ::= <identifier>
<subscripted variable> ::= <array identifier>
               [<subscript list>]
<array identifier> ::= <identifier>
<subscript list > ::= <arithmetic expression>|
               <subscript list>, <arithmetic expression>
```

<u>Expression</u>

```
<arithmetic expression> ::= <primary> | <arithmetic expression>
               <aritmetic operator> <primary>
<primary> ::= <number>|<identifier> | <function>|
              (<arithmetic expression>)
<function> ::= <function designator> <primary>
<Boolean expression> ::= <Boolean primary>|
               <Boolean expression>∧<Boolean primary>|
               <Boolean expression>∨<Boolean primary>
<Boolean primary> ::= <truth value> | <variable> | <relation>|
               <Boolean primary>|(<Boolean expression>)
<relation> ::= <arithmetic expression> <relational operator>
              < arithmetic expression>
```

The syntax of the arithmetic and Boolean expressions -- in contrast with the ALGOL syntax -- does not specify the order of the execution of operations. But this is merely a question of interpretation, so we will discuss it in the course of semantics. The formal correctness of expressions is sufficiently defined by the syntax above.

Labels and switches

<label> ::= <identifier>

<switch expression> ::= <switch identifier>
[<arithmetic expression>]

<switch identifier> ::= <identifier>

Statements

<basic statement> ::= <assignment statement>|<copy statement>
<control statement>|<dummy statement>
<input statement>|<output statement>

<assignment statement> ::= <variable> := <arithmetic expression>|
<variable> := <Boolean expression>

<copy statement> ::= <variable>⇐<operand>

<control statement> ::= go to <label>| go to
<switch expression>| jump <label>| exit | stop

<dummy statement> ::=

<input statement> ::= read (<variable>) | input (<variable>)

<output statement> ::= print (<arithmetic expression>)|output
(<arithmetic expression>)

<unconditional statement> ::= <basic statement>|
<compound statement>

<conditional statement> ::= if <Boolean expression> then
<unconditional statement>| if <Boolean expression>
then <unconditional statement> else
<unlabelled statement>

<for statement> ::= for <simple variable> :=
<arithmetic expression> step <arithmetic expression>
until <arithmetic expression> do
<unlabelled statement>

<unlabelled statement> ::= <unconditional statement>|
<conditional statement>|<for statement>

<unlabelled statement string> ::= <unlabelled statement>|

<unlabelled statement string> ;
<unlabelled statement>

<compound statement> ::= **begin** <unlabelled statement string>
end

<statement> ::= <unlabelled statement> | <label> :
<unlabelled statement>

According to these definitions each statement may have no more than one label and no label may occur within a **for** statement or conditional statement or compound statement. Instead of procedure statement we have special statements for subroutine call and for return from subroutine /**jump**, **exit**/. Four special statements serve for input-output operations.

Program

<program> ::= <statement> ;|<declaration> ; <program>|<program>;
<statement> ;

It should be made clear that this formal system of syntactic definitions is incomplete. In fact, we have several restricitions concerning the kinds and types of identifiers ocurring in the program, i.e. the declarations and the statements of a given program must correspond to each other more strictly than it is expressed by the syntactic rules above. Thus, for instance, an array identifier of a subscripted variable must occur in an array declaration in the appropriate position, and the number of subscript expressions of the subscripted variable must be equal to the number of subscript bounds in the corresponding declaration etc. On the contrary the interpreter will quite exactly specify the formal correctness of a program.

3. The semantics of L_{si}

As stated in the introduction the complete semantics of the language will be given through the interpreter. This means that we must define here only the meaning of the constructions involved in the interpreter. Now we may refer to ALGOL-60 in the case of constituents which are also allowed in that language. Thus, for example, the meaning of arithmic expressions and of assignment statements etc. is borrowed from ALGOL-60. We have to explain here only the meaning of the elements, that are not used in ALGOL-60.

First we define the basic symbols ent and stand, which represent two standard functions. The expression ent x designates the integer number entier (x+0.5), i.e. the function ent x converts the value of x, where x is given in floating point representation, into the nearest integer number in fixed point representation. The function stand a converts reversely the integer number a into a normalized floating point number, which is equal to a.

The two basic symbols ent and stand are treated as operators with one argument. The dynamic sequence of operations is determined by priority rules just like in ALGOL-60. We have here the following order of priority:

1. ent, stand
2. ↑
3. ×, /
4. +, -
5. <, > , =, ≠
6. ¬
7. ∧
8. ∨

The input-output operations must relate to the set of external symbols available for the input-output mechanisms. The standard procedures denoted by basic symbols input and output deal with only one external symbol at a time. Every external symbol will be transformed by the procedure input into an integer number and reversely, each of these numbers will furnish the corresponding external symbol through the procedure output. This correspondence between external symbols and integer numbers need not be specified uniformly. We simply suppose that we have standard procedures of those types, and it is very easy indeed to write such subroutines for every computer.

Provided that we have specified a one-to-one correspondence between the external symbols and a subset of integer numbers, the effects of these procedures may be defined as follows: The statement of the form input (a) will make the computer read the next symbol from the input channel and the value of the corresponding integer number will be assigned to the variable a. The statement of the form output (A) , where A denotes an arithmetic expression, will make the computer print out the external symbol corresponding to the value of A as the next symbol in the output channel.

These two standard procedures would be sufficient for any input-output procedure. Nevertheless we have included two more statements in our language in order to keep simple the programming of arithmetic problems. These statements have the forms reed (x) and print (A) and perform the input and output of decimal numbers, respectively. The statement read (x) will make the computer read the next number from the imput channel and assign its value to the variable x. This operation treats every external symbol other than digit or decimal point as a skip symbol, and any skip symbol may serve as a separator of numbers. The statement print (A) will cause the computer to print out the value of expression A in decimal form. We might reduce these two statements to the previous ones which are more elementary but the details of such a reduction are of little importance.

The call for a subroutine will be expressed by a statement of the form jump label. This means that the normal sequence of the obeying of statements will be broken in the same manner as in the case of a go to statement (see in [5]), but a reference to the breaking point will be stored in a special push-down store. The statement exit will then effect a return to the place which was stored in the topmost position of that push-down store. This dynamic organization of subroutine jumps makes it possible that our subroutines may have more than one entry and also many exits. But the beginning and the end of a subroutine are not statically fixed at all, since we may have a subroutine jump to any label and several go to statements may follow before we meet an exit statement. An exit statement with an empty push-down store will be regarded as an error but otherwise it has a well-defined effect. This method of interpretation allows recursive calls for every subroutine without any trouble. The programmer, of course, must keep track of subsequent calls and exits in his program.

So far we have given the a priori definition of L_{SI} though, to a considerable extent by analogy with ALGOL 60. On the basis of the a priori syntax and semantics we may now describe the interpreter. Before doing so, let us recall the defining rule of the interpreter. It is a question, of course, whether the a priori and the a posteriori definitions are equivalent. But the same problem arises in connection with every ALGOL compiler. And since the standard definition of ALGOL-60 is insufficiently formalized, this problem could not be solved in a completely exact way. Furthermore the ALGOL semantics still, contain ambiguities, so the interpretations of ALGOL texts by different compilers may be and are, in fact, different. That is why a compiler is required by some experts as a means of an exact language definition. It is worth quoting E. W. Dijkstra's ideas about these questions (p. 34. in [6]).

"As the aim of a programming language is to describe processes, I regard the definition of its semantics as the design, the description of a machine that has as reaction to an arbitrary process description in this language the actual execution of this process. One could also give the semantic definition of the language by stating all the rules according to which one could execute a process, given its description in the language."... "I would rather use the metaphor of the machine that by its very structure defines the semantics of the language. In the design of a language this concept of the defining machine should help us to ensure the unambiguity of semantic interpretation of texts."...

"Furthermore, we should be prepared to face the fact that our defining machine will become incredibily unpractical and unrealistic: it will be so wasteful of storage space and number of operations that it will hurt the eyes of every honest programmer. For, in how far does our defining machine differ from a real one that is provided with a good translator? This translator probably demands thousands of instructions and is therefore scarcely a realistic proposition as language definition. We should realize, however, that the size of the translator is largely due to the fact that the process has to be carried out as efficiently as possible /and furthermore by a machine not specially designed for this language/. By disregarding all efficiency requirements and tailoring the machine to the language we can obtain a much simpler organization, so simple in fact, that it can very well be used as a means of language definition. /This must be possible; otherwise, how could we, poor humans, ever master the language?/"[3]

According to these ideas every programming language should have a defining machine specially designed for this language. The question is now, how to describe this machine. We have chosen for this purpose the same language we want to define. This method of language definition could be applied to every formal language and helps us to be precise in each case. At the same time one may make reductions with the aid of the interpreter by reducing the complex structures of the language to the more elementary ones. /See e.g. the possibility of reducing the read statement to a subroutine using only the input statement etc./ In other words the interpreter could be

---

[3]No construction of such a machine is given in Dijkstra's paper.

written in a subset of the language.

The use of the same language for describing the interpreter has a special advantage in automatically testing the correctness of the interpretation. Provided that one has translated the interpreter into a machine language this machine will be able to obey any programm written in L_{SI}. Since the interpreter is written in the same language, it could also be interpreted. Thus, the reiterated interpretation of any program must produce the same result as its simple interpretation, if the man-made translation of the interpreter was correct.

4. The description of the interpreter.

Before describing the interpreter in terms of L_{SI}, we first present its broad outlines. Because of the arithmetic character of the semantics of L_{SI} all the basic symbols and the operands must be coded somehow, in order to enable the interpreter to deal with programms, i.e. sequences of symbols. Thus the working method of the interpreter is based on the arithmetic of integer numbers. The codings of basic symbols used by the interpreter is shown in Table 1.

The coding of operands is a little bit complicated. We must remember that we are dealing with an internal representation of the language.

We suppose that we have a special input program which reads first every program to be interpreted and puts it into the core storage. Meanwhile this program performs a preliminary tranformation that yields the internal coding of the program. The external interpretation depends on the input facilities and is not very important. The preliminary transformation converts the external representation of each basic symbol into the corresponding code according to Table 1.

Table 1

basic symbol	code	basic symbol	code	basic symbol	code
integer	1	¬	17	:=	32
real	2	∧	18	⇐	33
Boolean	3	∨	19	read	34
common	4	(	20	print	35
switch	5	ent	21	input	36
array	6	stand	22	output	37
[	7	if	23	,	38
+	8	then	24	)	39
−	9	begin	25	exit	40
×	10	for	26	;	41
/	11	step	27	:	42
↑	12	until	28	end	43
<	13	do	29	stop	44
>	14	go to	30	else	45
=	15	jump	31	]	46
≠	16				

The external representation of operands must be different from the external representation of any basic symbol. The internal codes of operands supplied by the input program will also be different from the internal codes of basic symbols. Also the three categories of the operands must be kept strictly apart.

The truth values true and false will have the internal codes 101 and 103. The number 0 in fixed point representation will have the internal code 105 and all the numbers occuring in the given program will receive an odd number beginning from 107. In fact the input program builds up a reference table of operands where each operand will have two registers. The internal code of an operand will be equal to the serial number of the first register corresponding to the given operand in that table increased by 100. In this way the codes of operands are bound to be different from those of basic symbols. The size of the reference table of operands is naturally limited by the storage capacity of the computer.

The third category of operands, i.e. the identifiers will be coded by means of their enumeration within the given program just as in the case of numbers, but the assignement of registers from the reference table will be carried out from the upper limit backwards.

Thus the input of the program will yield a sequence of internal codes /basic symbols and operands/, i.e. the program in an internal representation and a reference table of operands. Beside this the numbers /i.e. the constans of the program/ will be converted into their machine internal equivalents and stored in the data area of the storage.

The reference table of operands will contain two informations about each operand. The first register assigned to an operand will hold its type /see later/, while the second holds a pointer to the storage location, where the quantity represented by the operand will be stored. The types of numbers /real or integer/ can be recognized already by the input program /according to the presence or absence of a decimal point/, and storage locations containing their values can also be assigned to them. But the types and locations of identifiers can be established only on the basis of the declarations, which will be processed by the interpreter and not by the input program. So the input program does not provide these entries for the identifiers but uses their registers in the reference table for storing their external representation for enabling their identification at any place of their occurence. /i.e. it produces an identifier list and replaces every occurence of an identifier in the program by its serial number in that list increased by 100./

The input program carrying out this preliminary transformation can be written also in L_{SI} and we have done so, in fact, using a concrete external representation with five channel paper tape code. The external representation does not form a substantial part of the interpreter, therefore we have abandoned its detailed description in this paper.

The actual /real/ interpreter consists of two parts. The first part is the processing of the declarations while the second part is the interpretation and execution of the statements. These two parts form a complete program, which we will describe in a reference form of L_{SI}, where we use an ALGOL-like representation of the identifiers. A one-to-one correspondence between this reference language and the internal representation /or any external representation/ of the language must be guaranteed. This can be easily shown in our case.

Throughout the interpreter we shall use some simple variables that may be considered special registers of the defining machine. These are:

r	sequence-control register
a,b	registers for operands of type integer
x,y	registers for operands of type real
d,e	registers for operands of type Boolean
u	register for operands of type common
c	label register
g	type register

In addition to these variables we give a list of the most important subscripted variables, some of which are push-down lists. /See Table 2./

The Greek letters denote here and everywhere in the interpreter integer numbers that ought to be given explicitly, but they are limited only by the capacity of the computer. We may consider them as program parameters and we need not fix them once and for all.

Table 2

Subscripted variable	Subscript bound	Comment
P [r]	μ	Program
S [t]	τ	Push-down list of basic symbols
Q [v]	α	Push-down list of operands
W [i]	β	Reference table of operands
E [f]	φ	Directory table
I [h]	χ	Push-down list of subscript depth
Z [q]	ψ	Push-down list of cycle depth
U [p]	π	Push-down list of subroutine depth
A [n]	ν	Data area

The interpreter program begins with the following declarations:

<u>integer</u> r, n, a, b, i, j, k, l, R, N, G, f, dim, type, c, g, m, s, t, v, h, p, q, w;

<u>real</u> x, y; <u>Boolean</u> d, e; <u>common</u> u;

<u>integer array</u> P[μ], W[β], E[φ], S[τ], Q[α], I[χ], Z(ψ), U[π];

<u>common array</u> A[ν]; <u>swith</u> K [...

The labels of switch K are shown in a separate table /see Table 3./ The input program should follow just after the declarations, but as we said before, we do not want to describe it here, so we sum up its main effects only.

The preliminary transformation made by the input program places the program to be interpreted in the array P, and the index of the last element of P occupied by the program read-in will be assigned to the variable R. The input program will reserve pairs of registers in the reference table for the operands. The truth values and the number zero will always correspond to the first six elements of W. The subsequent elements of W /from W [7] to W[ℓ]/ will correspond to the numbers, while the last elements /from W [k] to W[β]/ to the identifiers occurring in the given program. Thus the variables ℓ and k will represent the limits of the occupied parts of the table W. /If $\ell \geq$ k then we have storage overflow./

The truth values and the number zero /in machine intern forms/ will be placed as A[1], A[2] and A[3], respectively. All the numbers will also be placed in the data area A and the value of the variable n will represent the total number of occupied elements from A. The type of each number and its place index in A will be stored also in W. We are denoting the types also by integer numbers as follows in Table 4.

Table 4.

integer 1	real 2	Boolean 3	common 4	label 5
integer array 6	real array 7	Boolean array 8	common array 9	switch 10

Thus, for example, if the first number in a given program is 1966, then we shall have W[7] = 1, W[S] = 4 and A[4] = 1966, and if the next number is 3.14 then W[9] = 2, W[10] = 5 and A[5] = 3.14 etc. /these two numbers will be replaced in P by 107 and 109 respectively at any place they occur./

Having finished the reading-in of the program and the preliminary transformation the input program will switch over to the processing of declarations. This part of the interpreter could be written as follows:

```
decl:     r:= 0;
          for i:= ℓ step 1 until β do  W[i] := 0;
new:      jump rstep; if  P[r] > 5  then go to   obey;
          if P[r] = 5 then go to  swit else  typ:= P[r] ;
list:     jump rstep; if P[r] = 6 then go to arr;
more:     jump take;
          if n < ν then n:= n+1 else go to oflow;
          W[j] := typ; W[j+1] := n; jump  rstep;
          if P[r] = 38 then jump rstep else if P[r] = 41 then
             go to new else go to error;
          go to more;

rstep:    if r < R then r:= r+1 else go to error; exit;
take:     j:= P[r] -100; if j < ℓ ∨ W[j] ≠ 0 then go to error;
          exit;

arr:      typ:= typ + 5;
repeat:   jump rstep; jump take;
          if f < φ then f:= f+1  else go to oflow;
          W[j] := typ; W[j+1] := f;
          dim:=0; G:=1; jump rstep;
          if P[r] ≠ 7 then go to error;
cycl:     jump rstep; i:= P[r]-100;
          if i < 7 ∨ i > k ∨ W[i] ≠ 1 then go to error;
          dim:= dim+1;
          if f+1+dim > φ then go to oflow;
          E [f+1+dim] ⇐ A[W [i+1] ]; G:=G × E[f+1+dim] ;
          jump rstep; if P[r]=38 then go to cycl;
          if P [r] = 46 then go to last else go to error;
last:     if n+G > ν then go to  oflow;
          E [f] := dim; E [f+1] := n+1;
          f:=f+1+dim; n:= n+G;
          jump rstep; if P[r] = 38 then go to repeat;
          if P [r]= 41 then go to new;
```

```
swit:     jump rstep; jump take;
          if f<φ then f:= f+1 else go to oflow;
          W[j] := 10; W[j+1]:= f; G:=0;
          jump rstep; if P[r]≠ 7 then go to error;
again:    jump rstep; i:= P[r]-100; G:= G+1;
          if i < ℓ ∨ (W[i]≠0 ∧ W [i] ≠5) then go to error;
          if  n+G > ν then go to oflow;
          W [i] := 5; A [n+G] ⇐ i;
          jump rstep;  if P[r]= 38 then go to again;
          if  P[r]= 46  then go to final else go to error;
final:    if  f+2>φ then go to oflow;;
          E [f] :=1;  E [f+1] :=n+1;  E [f+2] :=G;
          f:=f+2;  n:=n+G; jump rstep;
          if  P [r] = 41 then go to new else go to error;
```

This section of the interpreter clears first the upper part of the reference table W. Then decides at the label "new", whether a declaration or the first statement of the program comes next. /The declarations begin with basic symbols, the codes of which are less than 5. See Table 1./ If the symbol of type is followed by the symbol <u>array</u>, then the value of "typ" must be increased by 5. /See Table 4./ The type declarations are processed at the label "more", while the array declarations at the label "arr". For each simple variable two elements of W will be reserved, the first of which will contain the type of the variable. In the data storage A an element will also be reserved for each simple variable, and the subscript of this element of A will be stored in the second element of W corresponding to the variable. /Pointer./ During this processing of declarations several checkings are made to find out whether the declarations are consistent or not. In the latter case the program switches over to the label "error".[4]

For the arrays and switches a directory table E is used. The pointer in the second element of W corresponding to these identifiers will refer to an element of E. Here in E /in its next free element E [f] / the dimension of the array will be stored. The subsequent element of E/E [f+1]/ will show the element of A, which corresponds to the first element of the array. Just after these two elements of E there will be reserved one more element for each subscript bound. The number of subscript bounds given in the extension of the array /see in syntax, p. 4./ gives the dimension of the array. In the data area we have to reserve as many elements for each array as the product of the subscript bounds.

So far as the directory table E is concerned the switches are dealt with as one-dimensional arrays. The processing of switch declarations is made by the section of the interpreter starting at the label "swit". In the case of a switch values will be assigned, contrary to the arrays, to the corresponding elements of A as well since these represent labels in a given sequence. Each of these elements of A, therefore, will be a pointer showing the element of W, which corresponds to the identifier of that label. /The assignment of elements of W to identifiers was already effected by the input program./ The identifiers occurring in a switch declaration within the brackets must be labels, i.e. they must occur somewhere in the program before a : /colon/. During the processing of the declaration, however, we cannot be sure of this, but we fix their types as labels /5/. This is useful here for checking the consistency of declarations, i.e. to make sure that these identifiers should not be declared otherwise.

---

[4]The label "error" is used everywhere in the interpreter to express that a syntactic error has been detected in the program. In fact, we should have used different labels instead to distinguish the different kinds of errors.

After the settlement of declarations the interpreter will continue its work at the label "obey" with the initialization of the interpretation of statements. This part of the interpreter is controlled by the following few groups of statements, which cooperate with switch K. The symbols of the program will be processed one by one and each time an operand is encountered its code is decreased by 100 and is stored in Q. Each basic symbol will be matched against the previous one as to determine the further steps with the aid of switch K.

```
obey:      t:=1; S [1] :=0; v:=0; m:=k; N:=n;
           h:=0-1; q:=0-1; p:=0-2; go to here;

step:      jump rstep;
here:      if P [r]>100 then go to stack;
           if P [r]<7 ∨ P [r] > 46 then go to error;
iter:      s:=S [t] × 40+P [r]-6; go to K [s];

stack:     if v < α then v:=v+1 else go to oflow;
           Q [v] :=P [r] -100; go to step;

pd:        if t < π then t:=t+1 else go to oflow;
replace:   S [t] :=P [r] -6; go to step;

red:       t:=t-1; go to iter;

extin:     t:=t-1; go to step;
```

At the label "here" the actual element of P is examined. The label "pd" represents the push-down operation in the push-down list of basic symbols S, while "replace" describes the replacement of its topmost element by the actual element of P and "red" the reduction of S. The label "extin" symbolizes that the topmost element of S and the actual element of P extinguish each other.

In order to make it easy to follow the working method of the interpreter we first describe the most important subroutines. The following subroutine picks up two elements from the top of Q and puts the values of the corresponding operands in the appropriate registers of the defining machine. It may happen that one or both of these operands are working cells. The working cells required by the execution of the program are identified in the same manner as the declared variables, but the elements of W assigned to them must have subscripts between k and ℓ. In the data area A the elements representing working cells will have subscripts greater than N. It may be that we need a " pseudo working cell", i.e., a pair of elements of W specifying a selected element of an array or switch. We do so in each case when subscripted variable or a switch expression must be processed. The subroutine "oper" will clear the working cells which eventually correspond to the operands used up.

```
oper:   if v < 2 then go to error;
        i:=Q [v] ; j:=Q[v-1]; v:=v-2;
        if W [i] =1 ∧ W [j] =1 then go to int;
        if W [i]<3 ∧ W[j]<3 then go to rea;
        if W [i]=3 ∧ W[j] =3 then go to bool;
        go to error;
int:    a⇐A [W[i+1]]; b⇐A [W[j+1]]; g:=1; go to clr2;
rea:    if W[i]=1 then begin a⇐A [W[i+1]] ; x:=stand a end
            else x⇐A [W[j+1]];
        if W[j]=1 then begin b⇐A [W[j+1]]; y:= stand b end
            else y⇐A [W[j+1]]; g:=2; go to clr2;
clr2:   if W[i+1]=n∧n > N then n:=n-1; if k < i∧i < ℓ then
            begin if i=m then m:=m-2 else go to error end;
clr1:   if W[j+1] =n ∧n > N then n:=n-1, if k < j∧j < ℓ then
            begin if j=m then m:=m-2 else go to error end;
        exit;
```

The following subroutine is similar to the first, but it deals with only one operand.

```
value:     if v <1  then go to error; j:=Q [v]; v:=v-1;
           if W[j] > 4 then go to error else g:=W[j];
           if g=1 then  a⇐A [W[j+1]] else if g=2 then
                 x⇐A[W[j+1]] else if g=3 then d⇐[A W[j+1]]
                    else if g=4 then u⇐A[W[j+1]];
           go to clrl;
```

The task of the next subroutine is the opposite to that of the above, since it stores a value from one of the distinguished registers of the machine into a new working cell and keeps track of it in the topmost element of Q.

```
store:     if  v<α then v:=v+1 else go to oflow;
           if m+3<ℓ then m:=m+2 else go to oflow;
           if n<ν then n:=n+1 else go to oflow;
           Q[v]:=m; W[m]:=g; W[m+1]:=n; w:=n;
put:       if g=1 then A[w]⇐a else if g=2 then A[w]⇐x else
              if g=3 then A[w]⇐d else if g=4 then A[w]⇐u
                    else go to error;
           exit;
```

The subroutine "store" is used /called/ by the next one, which assigns the value of the operand at the top of Q to the operand just below it in Q. Both of the operands will be thrown out, thereafter, from Q.

```
putaway:   jump value; if v <1 then go to error;
           j:=Q[v]; v:=v-1;
           if W[j]>4 then go to error;
           if g=1∧W[j]=2 then begin g:=2; x:= stand a end else
                 if g=2∧W[j]=1 then begin g:=1;
                       a:= ent (x+0.5) end else if g≠W[j]∧g≠4∧
                             W[j]≠4 then go to error;
           w:=W[j+1]; jump put; jump clrl ;
           exit;
```

Now we have to describe those parts of the interpreter which are designed for selecting a given element of an array or of a switch through the evaluation of subscript expressions. First we give this part, which begins at the label "brack" and will be activated whenever a left bracket is encountered. The push-down list of subscript depth will be increased by two elements (I[h] , I[h+1]).

```
brack:    if v < 1  then go to error;
          if W[Q[v]]<6 then go to error;
          if h+2 <χ then h:=h+2 else go to oflow;
          I[h]:=0; I[h+1]:=0; go to pd;
```

The following passage will compute the subscript increment. The operand of the subscript expression is supposed to be evaluated and entered at the top of Q. The array or switch identifier is registered in Q next to the top. A part of the work will be done by the subroutine "ind".

```
incr:     jump ind;
          if ¬ I[h]<E[f] then go to error;
          I[h+1]:= (I[h+1] +a-1) × E [f+2+I[h]] ;
          go to step;

ind:      jump value; if g=2 then a:=ent (x+0.5) ;
          if g > 2  then go to error; g:=1;
          if v < 1  then go to error;
          f:= W[Q[v]+1] ;  I [h]: = I[h]+ 1;
          if a < 1  then go to error;
          if a > E [f+1+I[h]] then go to error;
          exit;
```

When the left bracket matches the right one, the required element of the array or switch will be selected with the aid of the following passage of the interpreter, where also the subscript depth will be reduced. The code of the "pseudo working cell" referring directly to the selected element will stay at the top of Q.

```
elem:     jump ind; i:=Q[v] ; if  W[i]<6 then go to error;
          if I[h] ≠ E[f] then go to error;
          I[h+1]:=I[h+1]+a;
          if m+3<ℓ then m:=m+2 else go to oflow;
          Q[v]:=m; W[m]:=W[i]-5;
          W[m+1]:=E[f+1]+I[h+1];
          h:=h-2; go to extin;
```

In possession of the subroutines and passages of the interpreter described up to this point, we may easily give the statements for processing of arithmetic and logical operations and of relations. Since these are very similar, we shall give only some examples.

```
add:      jump oper; if g=1 then a:=b+a else if g=2 then
               x:=x+y else go to error; jump store; go to red;
exp:      jump oper; if g=1 then begin if a>0 then a:= b↑a
               else begin g:=2; x:= stand b↑stand a end end
               else if g=2 then x:=y↑x else go to error;
          jump store; go to red;

less:     jump oper; if g=1∧b<a∨g=2∧x<y  then d:=true
               else if    g>2 then go to error else d:=false;
          g:=3; jump store; go to red;
conj:     jump oper; if g≠3 then go to error;
               d:=d∧e; jump store; go to red;
```

The two standard functions and the negation are operations with one argument.

```
ent:      jump value; if g=2 then a:=ent x else if g>2 then
               go to error; g:=1; jump store; go to red;
neg:      jump value; if g≠3 then go to error;
               d:=¬d; jump store; go to red;
```

The interpretation of the assignement statement and that of the copy statement are very simple too. The common type is excluded from the assignement statement.

```
ass:    if W[Q[v]]=4 ∨ W[Q[v-1]]=4 then go to error;
        jump putaway; go to red;
copy:   jump putaway; go to red;
```

The input-output operations are a little more complicated, because the kinds and types of their parameters are in some of them rather restricted.

```
inp:    if v<1 then go to error;
        j:=Q[v]; v:=v-1;
        if W[j]=1 ∧ W[j+1]>k then input(a) else go to error;
        A[W[j+1]]⟸a; go to red;

out:    jump value; if g=1 then output (a) else
             go to error; go to red;

read:   if  v<1  then go to error;
        j:=Q[v]; g:=W[j]; v:=v-1;
        if W[j+1] < k then go to error;
        if g=1 then read (a) else if g=2
             then read (x) else go to error;
        jump put; go to red;

print:  jump value; if g=1 then print (a) else if g=2 then
             print (x) else if g=3 then print (d) else
             go to error;
        go to red;
```

According to the syntax of L_{SI} a label is an identifier, but it must occur somewhere in the program text followed by a colon. Each time a colon is encountered a pointer to the next symbol of the program /i.e. to the first symbol of the labelled statement/ will be recorded in W as the pointer connected with the identifier entered in Q just before. Thus, each pointer in W connected with an identifier of type label will represent a subscript of P. This work will be done by the following passages.

```
colon:      if v < 1 then go to error;
            j:=Q[v] ; v:=v-1;
            jump reg; go to step;

reg:        if(W[j]≠0 ∧ W[j]≠5) ∨ (W[j]=5 ∧ W[j+1]≠0 ∧ W[j+1]≠r+1)
                 then go to error;
            W[j]:=5; W[j+1]:=r+1; exit;
```

In order to be able to perform a control statement we have to find the statement that is labelled by a given label. If this label has already a pointer recorded in W, then we can go immediately to that place, if not, then we must search for this label in the subsequent part of the program. It can easily shown that the reverse search is never needed. We give first the subroutine for this value /pointer/ assignement to the label.

```
point:      if v < 1  then go to error;
            i:=Q[v]; v:=v-1;
            if W[i]≠5 ∧ W[i]≠0  then go to error;
            if i > ℓ -1 then go to label;
            if i:=m then m:=m-2 else go to error;
            if W[i]=5 then i⇐[A W[i+1]] else go to error;
label:      if W[i]=5 ∧ W[i+1]≠0 then go to found;
srch:       jump rstep;  if P[r]≠42 then go to srch;
            j:= P[r-1]-100; if j<ℓ then go to error;
            jump reg; if j≠i then go to srch;
found:      r:=W[i+1]; exit;
```

Since no label may occur within a compound or conditional or <u>for</u> statement, no information must remain in the push-down lists S and Q, when a go to statement is executed. In other words any go to statement jumps out /up to the zero level/ from all compound structures embracing it. This is true, however, only under one condition, namely if it is not involved /in dynamic sense/ in a subroutine. To return from a subroutine is possible only through an exit statement. After an exit statement we must find the push-down lists S and Q in the same state as they were last time when a subroutine call was performed. These states are reserved in the push-down list of subroutine depth. Therefore a go to statement must restore these states if the push-down list of subroutine depth is not empty else S and Q must be cleared up to their bottoms. The passages concerned with control statements are as follows:

```
go:       jump point;
          if p>0 then begin t:=U[p+1]; v:=U[p+2] end
                else begin t:=1; v:=0 end;
          go to here;

call:     if p+4<π then p:= p+3 else go to oflow;
          U[p]:=r; U[p+1] :=t-1;
          jump point; U[p+2]:=v;
          go to here;

exit:     if p<0 then go to error;
          r:=U[p]; t:=U[p+1]; v:=U[p+2]; p:=p-3; go to here;

stop:     stop;
```

The conditional statement and the for statement have the common feature that according to given conditions some parts of them must be neglected. These jumps are essentially differen from those of the control statements, because here a search is always required. The for statement, however, has a real jump each time the loop is repeated. During the processing of these statements the interpreter will store different basic symbols in S according to the conditions evaluated. In case of matching the symbols <u>if</u> and <u>then</u> a decision is made. By that time the Boolean expression between them must have been evaluated and the corresponding operand holding its value is sitting at the top of Q. If this value is <u>true</u> then the symbol <u>if</u> will be replaced by the symbol <u>then</u>, else the sequence control register r will step forward until the corresponding <u>else</u> will be found. In the latter case the symbol <u>if</u> will be extinguished by the symbol <u>else</u>. The subroutine "skip" is called for whenever a compound statement is found in the segment to be neglected, for in this case the neglecting must not be stopped by a corresponding delimiter occuring within the <u>begin</u> and <u>end</u> symbols. And, since many compound statements may be embedded in one another, the subroutine "skip" can be called recursively from itself. The whole job with the conditional statements can be solved by the following four passages of the interpreter:

```
decide:   jump value ; if g≠3 then go to error;
          if d then go to replace else go to ignor;

ignor:    jump rstep;
          if P[r]=41 then go to extin;
          if P[r]=43 then go to red;
          if P[r]=45 then go to extin;
          if P[r]=25 then jump skip; go to ignor;

skip:     if r<R then r:=r+1 else go to error;
          if P[r]=45 then exit;
          if P[r]=25 then jump skip; go to skip;

enough:   jump rstep;
          if P[r]=41 then go to extin;
          if P[r]=43 then go to red;
          if P[r]=25 then jump skip; go to enough;
```

The push-down list of cycle depth is used for recording the controlled variable and the restart point of the cycle. The arithmetic expression representing the increment and the upper bound for the controlled variable are repeatedly evaluated in every iteration. The interpretation of the <u>for</u> statement is easy to follow throughout the following passages taking also into account, of course, the pervious parts of the interpreter and first of all switch K.

```
cycle:    if q+2 <ψ then q:=q+2 else go to oflow;
          if v<1 ∨ W[Q[v]]≠1 ∧ W[Q[v]]≠2 then go to error;
          Z[q]:=Q[v]; go to pd;

init:     if t < τ then t:=t+1 else go to oflow;
          S[t]:=21; Z[q+1]:=r;

forth:    jump rstep;
          if P[r]≠29 then go to forth; go to pd;

loop:     if v+1 <α then v:=v+2 else go to oflow;
          Q[v-1] :=Z[q] ; Q[v]:=Z[q];
          r:=Z[q+1] ; go to forth;

set:      jump putaway; if v<α then v:=v+1 else go to oflow;
          Q[v]:=Z[q]; go to pd;

check:    jump value; if g≠3 then go to error;
          if d then begin q:=q-2; go to enough end;
          if t<τ then t:=t+1 else go to oflow;
          S[t]:=21; go to pd;
```

With this we have finished the description of the interpreter.

5. Conclusion

The structure of the interpreter described above is based on the well-known method of push-down lists developed by F. Bauer and K. Samelson [2] .

On the other hand, the idea of a digital computer, which can be programmed in a mathematical language as suggested by L.

definition of the addition of integer numbers.
Kalmár [1] , formed the starting point of this work. It was assumed that this computer could be simulated on a normal digital computer. The program of simulation forms an interpreter of the mathematical formula language, which would be the machine language of the suggested computer. It seems useful to construct the program of simulation /i.e. the interpreter/ independently of the special machines available at the moment. Therefore we have chosen a universal language for describing our interpreter. But L_{SI} and the language, suggested by L. Kalmár are considerably different. Consequently the structure of our interpreter is even more different from that of Kalmár's computer. Thus, for example no putting on of redundant parantheses is required in L_{SI} etc. A more important feature is the recursive subroutine jump in L_{SI}. The dynamic interpretation of subroutine jumps and exits used by us is in our opinion very suitable also for building it into any computer. The interpreter as a whole can be also employed as the logical design of a special computer. One can use it, as we suggested in the introduction, in "teaching" our language to any computer.

The disadvantage of the interpretation technique lies in the necessity to follow the dynamic order of processing the statements. This has imposed the hardest restrictions on the language. A multipass translator such as e.g. the GIER ALGOL Compiler consisting of 9 passes [12] , can gather all the necessary informations from the program context in order to work as efficiently as possible. This concerns first of all the block structure and the concept of procedures of ALGOL.

Nevertheless, we suggest that the interpreter approach is rathet useful in formal language definition. Every formal language must have a subset which could be used as a means of its precise definition. It is up to us how far we want to go in reducing the complex structures to the more elementary ones. Theoretically we may describe the algorithm of the interpretation as a Turing - machine, which yields even an exact semantic

Returning to our interpreter we may summarize our results in the following theorems.

Theorem 1. To the formal language L_{SI} an interpreter written in the same language can be given such that describes the execution of any program written in this language.

Theorem 2. The language defined by the syntax given in paragraph 2 constitutes a subset of the language accepted by the interpreter.

This second theorem is not reversible, because we can construct expressions which are not allowed by the syntax given in paragraph 2 and can still be accepted by the interpreter. This is because of using different push-down lists for basic symbols and for operands, so the expression $a \times b(-c)$ will be interpreted as if it were $a \times (b-c)$. But any correct expression in the sense of paragraph 2 can be accepted by the interpreter, i.e. it will not cause the interpreter to perform a statement of the form go to error. This discrepancy in the two levels of syntactic definitions could be repaired, but we may accept only the second as a complete definition. The interpreter itself is written correctly in both senses.

Finally, we should like to remark, that the way of interpretation used here could be applied with appropriate modifications also in the design of translators.

Table 3

P[r] / S[t]	7 [	8 +	9 −	10 ×	11 /	12 ↑	13 <	14 >	15 =
6	brack	!	!	!	!	!	!	!	!
7 [	brack	pd	pd	pd	pd	pd	!	!	!
8 +	brack	add	add	pd	pd	pd	add	add	add
9 −	brack	sub	sub	pd	pd	pd	sub	sub	sub
10 ×	brack	mult	mult	mult	mult	pd	mult	mult	mult
11 /	brack	div	div	div	div	pd	div	div	div
12 ↑	brack	exp	exp	exp	exp	exp	exp	exp	exp
13 <	brack	pd	pd	pd	pd	pd	!	!	!
14 >	brack	pd	pd	pd	pd	pd	!	!	!
15 =	brack	pd	pd	pd	pd	pd	!	!	!
16 ≠	brack	pd	pd	pd	pd	pd	!	!	!
17 ¬	brack	pd	pd	pd	pd	pd	pd	pd	pd
18 ∧	brack	pd	pd	pd	pd	pd	pd	pd	pd
19 ∨	brack	pd	pd	pd	pd	pd	pd	pd	pd
20 (	brack	pd	pd	pd	pd	pd	pd	pd	pd
21 <u>ent</u>	brack	ent	ent	ent	ent	ent	ent	ent	ent
22 <u>stand</u>	brack	stan	stan	stan	stan	stan	stan	stan	stan
23 <u>if</u>	brack	pd	pd	pd	pd	pd	pd	pd	pd
24 <u>then</u>	brack	!	!	!	!	!	!	!	!
25 <u>begin</u>	brack	!	!	!	!	!	!	!	!
26 <u>for</u>	brack	!	!	!	!	!	!	!	!
27 <u>step</u>	brack	pd	pd	pd	pd	pd	!	!	!
28 <u>until</u>	brack	pd	pd	pd	pd	pd	!	!	!
29 <u>do</u>	brack	!	!	!	!	!	!	!	!
30 <u>go to</u>	brack	!	!	!	!	!	!	!	!
31 <u>jump</u>	!	!	!	!	!	!	!	!	!
32 :=	brack	pd	pd	pd	pd	pd	pd	pd	pd
33 ⇐	brack	!	!	!	!	!	!	!	!
34 <u>read</u>	!	!	!	!	!	!	!	!	!
35 <u>print</u>	!	!	!	!	!	!	!	!	!
36 <u>input</u>	!	!	!	!	!	!	!	!	!
37 <u>output</u>	!	!	!	!	!	!	!	!	!

Remark: The label „error" is represented here by the symbol „!".

16	17	18	19	20	21	22	23	24	25	26
≠	¬	∧	∨	(	<u>ent</u>	<u>stand</u>	<u>if</u>	<u>then</u>	<u>begin</u>	<u>for</u>
!	!	!	!	!	!	!	pd	!	pd	pd
!	!	!	!	pd	pd	pd	!	!	!	!
add	!	add	add	pd	pd	pd	!	add	!	!
sub	!	sub	sub	pd	pd	pd	!	sub	!	!
mult	!	mult	mult	pd	pd	pd	!	mult	!	!
div	!	div	div	pd	pd	pd	!	div	!	!
exp	!	exp	exp	pd	pd	pd	!	exp	!	!
!	!	less	less	pd	pd	pd	!	less	!	!
!	!	gr	gr	pd	pd	pd	!	gr	!	!
!	!	eq	eq	pd	pd	pd	!	eq	!	!
!	!	uneq	uneq	pd	pd	pd	!	uneq	!	!
pd	neg	neg	neg	pd	!	!	!	neg	!	!
pd	pd	conj	conj	pd	!	!	!	conj	!	!
pd	pd	pd	disj	pd	!	!	!	disj	!	!
pd	pd	pd	pd	pd	pd	pd	!	!	!	!
ent	!	!	!	pd	ent	ent	!	!	!	!
stan	!	!	!	pd	stan	stan	!	!	!	!
pd	pd	pd	pd	pd	pd	pd	!	decide	!	!
!	!	!	!	!	!	!	!	!	pd	pd
!	!	!	!	!	!	!	pd	!	pd	pd
!	!	!	!	!	!	!	!	!	!	!
!	!	!	!	pd	pd	pd	!	!	!	!
!	!	!	!	pd	pd	pd	!	!	!	!
!	!	!	!	!	!	!	pd	!	pd	pd
!	!	!	!	!	!	!	!	!	!	!
!	!	!	!	!	!	!	!	!	!	!
pd	pd	pd	pd	pd	pd	pd	!	!	!	!
!	!	!	!	!	!	!	!	!	!	!
!	!	!	!	pd	!	!	!	!	!	!
!	!	!	!	pd	!	!	!	!	!	!
!	!	!	!	pd	!	!	!	!	!	!
!	!	!	!	pd	!	!	!	!	!	!

Table 3. /continued/

S[t] \ P[r]	27 step	28 until	29 do	30 go to	31 jump	32 :=	33 ⟸	34 read	35 print
6	!	!	!	pd	pd	pd	pd	pd	pd
7 [	!	!	!	!	!	!	!	!	!
8 +	add	add	add	!	!	!	!	!	!
9 −	sub	sub	sub	!	!	!	!	!	!
10 ×	mult	mult	mult	!	!	!	!	!	!
11 /	div	div	div	!	!	!	!	!	!
12 ↑	exp	exp	exp	!	!	!	!	!	!
13 <	!	!	!	!	!	!	!	!	!
14 >	!	!	!	!	!	!	!	!	!
15 =	!	!	!	!	!	!	!	!	!
16 ≠	!	!	!	!	!	!	!	!	!
17 ¬	!	!	!	!	!	!	!	!	!
18 ∧	!	!	!	!	!	!	!	!	!
19 ∨	!	!	!	!	!	!	!	!	!
20 (	!	!	!	!	!	!	!	!	!
21 ent	ent	ent	ent	!	!	!	!	!	!
22 stand	stan	stan	stan	!	!	!	!	!	!
23 if	!	!	!	!	!	!	!	!	!
24 then	!	!	!	pd	pd	pd	pd	pd	pd
25 begin	!	!	!	pd	pd	pd	pd	pd	pd
26 for	init	set	check	!	!	cycle	!	!	!
27 step	!	add	!	!	!	!	!	!	!
28 until	!	!	gr	!	!	!	!	!	!
29 do	!	!	!	pd	pd	pd	pd	pd	pd
30 go to	!	!	!	!	!	!	!	!	!
31 jump	!	!	!	!	!	!	!	!	!
32 :=	ass	!	!	!	!	!	!	!	!
33 ⟸	!	!	!	!	!	!	!	!	!
34 read	!	!	!	!	!	!	!	!	!
35 print	!	!	!	!	!	!	!	!	!
36 input	!	!	!	!	!	!	!	!	!
37 output	!	!	!	!	!	!	!	!	!

36 input	37 output	38 ,	39)	40 exit	41 ;	42 :	43 end	44 stop	45 else	46]
pd	pd	!	!	exit	step	colon	!	stop	!	!
!	!	incr	!	!	!	!	!	!	!	elem
!	!	add	add	!	add	!	add	!	add	add
!	!	sub	sub	!	sub	!	sub	!	sub	sub
!	!	mult	mult	!	mult	!	mult	!	mult	mult
!	!	div	div	!	div	!	div	!	div	div
!	!	exp	exp	!	exp	!	exp	!	exp	exp
!	!	!	less	!	less	!	less	!	less	!
!	!	!	gr	!	gr	!	gr	!	gr	!
!	!	!	eq	!	eq	!	eq	!	eq	!
!	!	!	uneq	!	uneq	!	uneq	!	uneq	!
!	!	!	neg	!	neg	!	neg	!	neg	!
!	!	!	conj	!	conj	!	conj	!	conj	!
!	!	!	disj	!	disj	!	disj	!	disj	!
!	!	!	extin	!	!	!	!	!	!	!
!	!	ent	ent	!	ent	!	ent	!	!	!
!	!	stan	stan	!	stan	!	stan	!	!	!
!	!	!	!	!	!	!	!	!	!	!
pd	pd	!	!	exit	extin	!	!	stop	enough	!
pd	pd	!	!	exit	step	!	extin	stop	!	!
!	!	!	!	!	!	!	!	!	!	!
!	!	!	!	!	!	!	!	!	!	!
!	!	!	!	!	!	!	!	!	!	!
pd	pd	!	!	exit	loop	!	loop	stop	!	!
!	!	!	!	!	go	!	go	!	go	!
!	!	!	!	!	call	!	call	!	call	!
!	!	!	!	!	ass	!	ass	!	ass	!
!	!	!	!	!	copy	!	copy	!	copy	!
!	!	!	!	!	read	!	read	!	read	!
!	!	!	!	!	print	!	print	!	print	!
!	!	!	!	!	inp	!	inp	!	inp	!
!	!	!	!	!	out	!	out	!	out	!

REFERENCES

[1]L. Kalmár: A practical infinitistic computer "Infinitistic Methods" Proceedings of the Symposium on Foundation of Mathematics, Warsaw 1959, 347-362. Über einen Rechenautomaten, der eine mathematische Sprache versteht, Z. angew. Math. Mechan. 40/1960/ T64-T65. On a digital computer which can be programmed in a mathematical formula language, II. Hung. Math. Congr. Vol. 2. section 5, 3-16. O cifrovoj vychislitel'noj mashine s programmirovaniem posredstvom formal'nogo matematicheskogo jazyka. Kiberneticheskij sbornik, Novaja serija 1., 1965, 215-226.

[2]K. Samelson - F. Bauer: Sequential formula translation. Comm. of the ACM 3 /1960/ 76-83.

[3]Alan J. Belbourne - John M. Pugmire: A small computer for the direct processing of FORTRAN statements, Computer journal 8 /1965/ 1, 24-27.

[4]Edgar T. Irons: A Syntax Directed Compiler for ALGOL 60. Comm. ACM 4 /1961/ 51-55.

[5]R. A. Brooker - I. R. MacCallum - D. Morris - J. S. Rohl: The Compiler Compiler. Annual Rewiew of Automatic progr. 3 /1963/ 229-275.

[6]E. W. Dijkstra: On the design of machine independent programming languages, Annual Rewiew of Automatic Progr. 3 /1963/ 27-42.

[7]Jan V. Carwick: The definition of programming languages by their compiler, IFIP Working Conference, Vienna 1964 September.

[8]A. A. Markov: Teorija algorifmov. Trudy matematicheskogo instituta SSSR, tom 42.

[9]M. Davis: Computability and Unsolvability /McGraw-Hill 1958/.

[10]P. Naur /ed./: Report on the Algorithmic Language ALGOL 60, Comm. ACM 3 /1960/ 5, 299-314.

[11]P. Naur /ed./: Revised Report on the Algorithmic Language ALGOL 60, Comm. ACM 6 /1963/ 1, 1-17.

[12]P. Naur: The design of the GIER ALGOL Compiler, BIT 3 /1963/ 2, 3, 124-140, 146-166.

[13]V. M. Glushkov - A. A. Letichevskij - A. A. Stognij : Vxodnoj jazyk vychislitel'noj mashiny dlja inzhenernyx raschetov, Kibernetika I., 1965, 74-82.

[14]S. S. Kaminin - Z. Z. Ljubimskij: Algoritmicheskij mashinoorientirovannyj jazyk, Vychislitel'nyj centr AN SSSR, Moscow, 1966.

TOWARDS A NEW SYSTEM OF AUTOMATIC ANALYSIS[1]

D. Varga

1. Preliminary remarks

In the Computing Centre of the Hungarian Academy of Sciences a new work group was formed in 1966 under the name of "Documentation Linguistic Group", partly continuing the research already begun and being pursued within the Mathematical Linguistic Group. The group concentrates its research activity on the elaboration of a new system of automatic analysis.

The system under elaboration strives to realize a more adequate procedure of analysis than it is customary in the field of the automatic translation[2] so that the results /not only the final but also the partial results obtained in the course of the elaborating process/ will, it is believed, provide a useful raw material for linguistic theory as well. At the same time we do not want to forget about the requirement of the practical usability of the system: we want to establish an improved apparatus for the better "co-operation" of man and machine. On the one hand, we are convinced that further development of linguistic theory requires the testing and improvement of the results achieved up till now, and mainly the collecting of material on a larger scale than it was done in the past, and that the only way to realize this aim is to use computers more intensively in the process of linguistic research[3]. On the other hand, it is more and more evident that the realization of the experimental tasks of automatic translation and documentation is not possible without taking into consideration the results of linguistic theory to the greatest extent. Automatic analysis is tipically the field in which theoretical and experimental research must go hand in hand.

As our intention is to process texts of greater length with the help of computers, we do not consider sufficient such a form of co-operation of man and machine in which linguistically previously prepared texts have to be fed in[4]. Analysis of texts without previous preparation affords, of course, a wider field for manifold interpretation - consequently it means a harder task for the analysing algorithm to meet.

The algorithm of theoretical pretensions must of course, be suitable for the establishing of every variant of analysis[5], but it is not all. We are aware of the fact that due to the insufficiency of our knowledge concerning language, the system of linguistic rules must be improved in the course of progress and made more inclusive in many respects. That is why it is a very important requirement that the algorithm should make possible linguistical "feedback", i.e. the direct use of the results of the analysis for the improvement of the rule system without essentially altering the structure of the system. At this point human activity seems indispensable to discover the insufficiencies of the rule system by evaluating the obtained results of analysis. The introduction of improvements /the elaboration of modified matrices of analysis etc./ can of course be automatized again to a large extent[6].

2. The grammar of the analysis

To determine the type of the grammar to be applied in the course of the analyzing process is a problem which is not sufficiently clarified in the literature. We have to emphasize that starting from the text requires raising problems in quite another way than is customary for those who deal with questions of generative grammar. That is:

2.1. The "grammar" used for the analysis is not necessarily a grammar in the strict sense of the word, because its primary aim is not to distinguish the sentences of a language from the non-sentences. In our view, the elaboration of a so called "recognition grammar" could not be realized at all in the case of natural languages /cf. the unsolvable problems emerging already in connection with the CF grammars[7]. In the case of a natural language the only task that can be regarded realizable is that if we examine a sentence of this language it will be formally defined its structure then. We do not make any restriction concerning the case when the input sequence does not belong to the sentences of the respective language /as we do not expect about the logical implication to explain in advance whether the statement to which it applies, is true or not/.

2.2. Until the semantic and intersentence /i.e. extending beyond the sentence boundary/ limitations are not cleared sufficiently - a good analyzing system could render a great service just in this very field - it seems justified to further weaken the existing requirements. We require of an analysis carried out on the basis of pure syntactical conditions only that it should contain the sytactically correct analysis /analyses/ of the given sentence as one /some/ of the several possible resultant analyses. /There can be, of course, more than one such structure/. The selection of syntactical structures cannot be carried out exclusively according to syntactic viewpoints. The rules themselves can be considered as those having a definitely syntactic character, while among the conditions of their application there are already a great number of lexical-semanticaly and intersentence /topic-comment etc./ limitations[8].

2.3. The division of the grammar applied on the analogy of the generative grammar into phrase structure /basis/ and transformation levels[9] is irreal here because of we start out from given texts. In this case the inverse transformation correspondig to the last transformation must be applied first to the given sentence /brought into existence by the application of some transformations/. after it the transformation corresponding to the last but one, and so on, until we reach the level of the basis. But the fact is that we have to carry out all this without knowing in advance the proper structure of the sentence. It seems evident, that

a/ isolated tranformations cannot be inverted in every case /e.g. in the case of deletion transformations/;

b/ the inverse transformations obtained by the inverting of transformations would also be transformations, the application of these seems to be senseless without at least a partial knowledge of the structure.

3. The phase of determining the syntactic „frame structure"

Instead of the formal inversion of the generative process refering to the single sentences we need such an approach to the problems where in the first phase of the analysis the final form of the sentence serves as a direct basis for the definition of the structure.

3.1. This structure cannot be identical with the deep structure of the sentence[10], nor need it contain every information about the surface structure[11]. The establishing of a syntactic frame seems to be necessary which explains the /syntactic/ possibilites of the joining of the syntactic units into a coherent system. If we remain on the level of syntax, there may occur more such frames than if we take into consideration all semantical and intersentence limitations as well. /Sentences can have more syntactic frame structure even if, according to the "total" analysis, the syntactic structure of the sentence is unambiguous./ The selection of these structures as well as the determining of the functions of the constituents within this syntactic frame will be carried out in a separate phase of the analysis[12] /which functions also as an inverse transformation level/.

So the analysis is carried out in two phases. To these can be added a third, a preparatory phase whose function is to produce /by the morphological analysis of the words/ the sequense of syntactic units necessary for the definition of the syntactic frame. A more detailed description of these phases can be found in the journal "Nauchno - Texnicheskaja Informacija"[13].

3.2. On the first /syntactic/ phase of the analysis we intentionally leave out certain information, not because we regard them unimportant, but because their separation does not essentially influence the determination of the frame structure. Besides, to use them according to their own importance is possible only in the possession of the results of the analysis of the first phase. Some kinds of these information are yet of syntactic, others of intersentence character completing the former on the basis of indications beyond the sentence limit.

The syntactic information left out of consideration allude to the compound /consisting of more than one parts/ syntactic units' inner interdependencies and have no role at all in the first phase in the formation of the hierarchization of the units. /These are those additional informations which are contained by the dependency grammar against "phrase structure" grammars./ Such a kind of information for example, concerns the main constituent in a compound syntactic unit /if such a one can at all be separated /, and the kind of relations which exist between the main constituent and the sub-constituents subordinated to the former. Their analysis of a deeper character cannot be realized on the separate syntactic level /for example, the relation of the noun in genitive to the head noun; cp. the problem of the subjective genitive and that of the objective genitive/ and regarding its essence the whole problem belongs to the lexical-semantical analysis based on it.

3.3. In the future we want to build a part of the unregarded semantical and intersentence information into the first phase of the analysis as limiting conditions of the application of syntactic rules. This will make it possible that we should prevent the formation of certain semantically or contextually non-correct syntactic structures and by this speed up the operation of the algorithm.

However, the greatest part of the semantical and intersentence information cannot be examined separately but only with reference to a certain system of the interdependent syntactic units. The expedient processing of these begins only when the first phase of the analysis has already been finished and we have at our disposal /even if not necessarily unambiguisly/ the syntactic structure of the sentence.[14]

3.4. We have one another essential remark to make concerning the recognition of the syntactic "frame structure". In the same way as we do not stick to the "one to one" correspondence when building up the initial sign sequence for the syntactic analysis, it would also be unnecessary in the further course of the analysis to explain the result of hierarchization in the form of one single syntactic unit. From the point of view of further hierarchization it is sometimes worthwhile to "individualize" certain syntactic characteristics, to express them by separate symbols.

3.4.1. This solution is a very suitable /and at the same time simple/ apparatus for the carrying over necessary information, from the units of a lower hierarchizational level to those of a higher one. At the same time it also makes it possible that when we do not need the information for the further analysis, it can be "dropped" by the applied rule. By this, rules become more flexible, make use of the relative importance concerning the formation of the structure of the different pieces of syntactic information.

This principle is rather suitable for taking into account the auxiliary apparatus concerning concord and government in the different languages only till it is essential for then to be built into the greater units. Up to that point on the other hand the fulfilment of the conditions concerning concord and government together with syntactic linking can be examined with the same mechanism and in the same way /which is the main advantage of the linearization applied in the preparatory phase/. In our system, for example, a unit consisting of a transitive verb with its object, is equivalent to an intrasitive verb, etc. This method is also suitable for the processing of the so-called non-continuous structures.

3,4.2. This solution means at the same time that we have got beyond the "context free" bounds, and a grammar constructed in this way will be of the same rank as any "context sensitive" grammar. We wish to remark that the mathematical apparatus we used would make possible also the further widening of grammar /but this does not seem necessary for the time being/. Since the algorithm of the first phase depends exclusively on the notion of the continous linking of units we have not to observe the restriction that the resulting symbol string should be shorter than that replaced by it. The restrictions concerning the characteristics of the grammar are consequently not contained by the algorithm itself but they are implicitly included in the presented structure of the linguistic rule system.[16]

4. The phase of determining the semantic deep structure

4.1. The function of this phase is that relying on the various kinds of information of the syntactic frame structure/s/ and on the lexicological data belonging to the lexical units it should determine

a/ which kinds of syntactic frame structures can be semantically realized,

b/ what kind of lexical-semantical deep structure/s/ can be corresponded to a given syntactic frame structure.

In order to accomplish this task we have to determine the partial tasks under b/, because the syntactically possible but semantically incorrect structures can be eliminated just by the realization of task b/.

4.2. Briefly we could divide the process of determining the deep structure as follows:

/i/ The lexical-semantical "filling in" of the syntactic structure.

/ii/ The revealing of the inner interdependence relations of the compound syntactic units.

/iii/ The formation of the proper government spheres /around the adjectives in comparative degree, verbs, nouns formed verbs, etc./.

/iv/ The division into interdependent sub-structures of the total structure.

/v/ The determination of the way the sub-structures are being linked to each other /relative dependent clause, etc./.

/vi/ The determination of the total government system of lexical-semantical units of the structure /on the basis of the dictionary/.

/vii/ The "filling in" of the lacking constituents of the government system, and the determination of the referential relations existing among the single constituents.

It way be observed that while in the determination of the syntactic frame structure the grammar applied stood close to the IC grammar, in the present phase of analysis we would strive to work out a structure that is built up rather on the principles of dependency grammar. This can be explained by the facts that, on the one hand,

a/ the transition is simpler from the linear sequence to the partly linear tree built up according to the principle of continuity /mainly if the sentence is composed of constituents several times embedded into each other/,

b/ the conditions, rules of syntactic linking /mainly in the case of rules with more than two constituent/ can be more easily and precisely explained,

c/ the surface structure of the compound sentence can thus be expressed in a more homogeneous way.

On the other hand

a/ the dependency tree with some basic relations of transformation is suitable for the homogeneous description of the sentence structure,

b/ it can explain such interrelations among the lexical-semantical units which the previous structure cannot,

c/ it makes possible the leaving out of the details unimportant from the point of view of the lexical-semantical deep structure and thus it is more suitable for the explanation of the interrelations of the deep structure.

4.3. When interpreting the deep structure we take for our basis the semantical conception of Melchuk-Zholkowsky[17] /which in some respects is similar to the conception of Weinreich[18]/. The conception of Melchuk-Zholkovsky does not consider the semantical deep structure of the sentence as a <u>supplement</u> to the syntactic structure but it <u>replaces</u> the syntactic structure with an independent semantic one.

Consequently, we deviate from Chomsky's conception of deep structure in that we do not consider the constituents of the deep structure to be those syntactic constituents and relations which contain the information referring to the realization of the surface structure only.

But at the same time we regard as organic parts of the deep structure those transformation relations which refer to the linking of the semantic units <u>within</u> the deep structure, consequently they exist as constituents of the "syntax of the deep structure". These transformation relations should not, of course, contain those concrete froms which concern the production of the surface structure.

The notion of the predicate function plays a central role in this concept. Essentially the semantic deep structure can be derived from the recursive linking of the predicate functions and its filling in with some further lexical elements. In connection with this it is rather important that certain "transformations" applied to the functions /nominalization, etc./ can be considered as if a variable of an implicit predicate function were explicitly expressed. For example the verb "persuade" can be considered as a predicate function with three variables, PERSUADE /x, y, z/ the arguments of which are: x the subject who persuades, y the person persuaded and z the action into which x persuades y. From this F/x,y,z/ function — independently of the concrete forms of linguistic realization — can be expressed the x,y,z variables of the function.

Let us denote these with N_1/F/, N_2/F/, and N_3/f/. When one of the variables of some other predicate function occurs as one of the arguments of some predicate function /but not the predicate function itself/, then the connection between them can be established by the proper N_1, N_2 or N_3 functions.

In a similar way therefore a functional and more differentiated interpretation of nominalization can be established on an abstract level in semantic research, which would greatly promote the clearer and more exact exposition of the notion of deep structure.

REFERENCES

[1]A more detailed Russian version of this arhicle will be published in the journal "Nauchno-Tetexnicheskaya Informatsiya".
The first enterprise of greater significance of this research team was the completion of the Hungarian version of the KWIC-index. In 1967 this group organized, in close co-operation with the National Technical Library and Dokumentation Centre, the first international symposium of the socialist countries in the field of automatic translation in Budapest, under the title MASHPEREVOD-67. The material of the symposium is going to be published in Russian in the first half of 1968.

[2]Cf. Botos, I. Voprosy adekvatnosti analiza, Symposium MASHPEREVOD-67 in Budapest, 1967. /forthcoming/.

[3]Cf. National Science Foundation Symposium on the current status of research at the Linguistics Research Center /The University of Texas/. Austin /Texas/, 1963. and

Hays, D. G. Research procedure in machine translation. Santa Monica /Calif./, RAND Corporation, 1961. RAND RM-2916-PR.

[4]Cf. Kulagina, O. S. Ob ispol'zovanii mashiny pri sostavlenii algoritmov analiza teksta. In: Problemy kibernetiki, 7, Moscow, 1962, Fizmatizdat.

[5]Cf. Kuno, S., Oettinger, A. G. Multiple-path syntactic analyzer. In: Mathematical linguistics and automatic translation. Report NSF-8. Cambridge /Mass./, 1963.

Plath, W. J. Multiple-path syntactic analysis of Russian. In: Mathematical linguistics and automatic translation. Report NSF-12. Cambridge /Mass./, 1963.

Nagao, M, Studies on language analysis procedure and character recognition. Kyoto University, 1965.

Lejkina, B. M., Nikitina, M. I., Otkupshchikova, S. J.,

Fitialov, S. J., Cejtin, S. G. Sistema avtomaticheskogo perevoda, razrabatyvaemaja v gruppe matematicheskoj lingvistiki VC LGU. Nauchno-texnicheskaja informacija, 1966. 1. 40-50.

Vakulovskaja, G. V., Kulagina, O. S. Ob odnom algoritme sintaksicheskogo analiza russkix tekstov. In: Problemy kibernetiki, 18. Moscow, 1967, Fizmatizdat.

[6]Cf. É. Szöllősy and Zh. Varga's works in Documentation Linguistic Group in the Computing Centre of the Hungarian Academy of Sciences.

[7]Cf. Gladkij, A. V. Algoritmicheskaja neraspoznavaemost' sushchestvennoj neopredelennosti kontekstno-svobodnyx jazykov. In: Algebra i logika, 4, Novosibirsk, 1965. 53-64. and

Landweber, P. S. Decision problem of phrase-structure grammars, IEE Trans, vol. EC-13, 1964. 354-462.

[8]Cf. Varga, D. Strategija mashinnogo perevoda. Symposium MASHPEREVOD-67 in Budapest, 1967. /forthcoming/.

[9]Cf. Chomsky, N. Syntactic structures, The Hague, 1957. and

Chomsky, N, Aspects of the theory of syntax. Cambridge /Mass./, 1965.

[10]Cf. Mel'chuk, I. A. Avtomaticheskij sintaksicheskij analiz, I. Novosibirsk, 1964. and

Iordanskaja, l. N. Avtomaticheskij sintaksicheskij analiz, II. Novosibirsk, 1967.

[11]Cf. Beleckij, M. I. Beskontekstnye i dominacionnye grammatiki i svjazannye s nimi algoritmicheskie problemy. Kibernetika, 4, Kiev, 1967, 90-97.

[12]Cf. B. Vauquois's report in Jerevan, 2nd Congress of MT, 1967.

[13]There is presented the sharping of the Dömölki's algorithm, Nauchno-texnicheskaja informacija, serija 2, 1968. 3.

Cf. Dömölki, B. An algorithm for syntactic analysis, Computational Linguistics III, Budapest, 1964, 29-46.
Cf. also the Borshchev's algorithm,
Borshchev, V. B., Jefimova, J. N. O sokrashchenii perebora pri sintaksicheskom analize, Nauchno-texnicheskaja informacija, Ser 2. 1967, 10. 27-33.

[14]In the process of human understanding these two phases are naturally not separated so strictly, that is, the syntactic and semantic processing are closely linked together. The brain's understanding proceeds, however, also on the basis of the recognized syntactic relations. The difference could perhaps be explained in this way: in the course of human processing the semantical the interpretation concerning certain /hypothetically/ recognized sub-structures may be proceeded before their final places and relations in the sentence would come to light. /Otherwise it would be hard to imagine the semantic interpretation of the Hungarian sentence with a depth of 29 /!/ shown in Computational Linguistics III.

[15]This interpretation remains, of course, hypothetical till a further analysis has proved the results. Maybe we are not mistaken if stating that the process of understanding can also be interpreted as a sequence of certain "transformations" which modify and link together the recognized semantic blocks.

[15]Cf. Varga, D. Yngve's hypothesis and some problems of the mechanical analysis, Computational Linguistics III, Budapest, 1964, 47-74.

[16]The algorithm itself can also be used for the generation of sign sequences if we build into it a proper and suitable limitation to stop the process /for example we separate the terminal and non-terminal signs and / or limit the length of the sign sequence/.

[17] Cf. Zholkovsky, A. K., Mel'chuk, I. A. O sisteme semanticheskogo sinteza. Problemy kibernetiki, 19. Moscow, 1967.

[18] Weinreich, U. Explorations in semantic theory. In: Current Trends in Linguistics 3, The Hague 1966, Mouton, 395-479.

R E V I E W

THE PRAGUE BULLETIN OF MATHEMATICAL LINGUISTICS
/Universita Karlova, Praha/

by

I. Szelezsán and M. Stein

The periodical "The Prague Bulletin of Mathematical Linguistics" is published by the Prague Charles University twice a year. The first issue came out in 1964.

The members of the editorial committee are: Kařel Berka, Pavel Novák, Petr Sgall /editor-in-chief/, and Marie Těšitelová.

The periodical -- as we can read in the foreword of the first issue -- reports on the research directed by Prague University in the field of mathematical linguistics and its application. The periodical contains the original articles of the researchers of the University and of some other research institutes as well as the English and Russian language summaries of papers written in Czech. In addition, there are short summaries at the end of each volume of works concerning the mentioned field of research.

This field of research is rather wide and affords great possibilities for the research workers. This statement is borne out by the papers reviewed here. The themes are rather diversified: besides treatises dealing with the problems of Czech linguistics there are also to be found articles on mathematical linguistics, and purely on mathematical problems. The periodical also includes dissertations published over a number of issues. This way of editing, in our view, throws difficulties in the way of understanding. The treatises are going to be reviewed in their sequence of publication, volume by volume.

The foreword of the first issue /1964/ expressed the views of the group in connection with some problems. There refers to:

/1/ the relation between algebraic and statistical linguistics:

In the case of linguistic research the primary aim of which is the examination of qualitative characteristics, the use of statistical data can also prove helpful. The study of quantitative characteristics nowever is only a precondition insufficient in itself to lay a solid foundation for qualitative analysis. The results and problems of statistical linguistics have not yet been thoroughly connected with the basic problems of theoretical linguistics. From the viewpoint of the future development of the science of linguistics the establishment of a closer connection between the statistical approach and the theoretical basis rendered by algebraic linguistics may well be of major importance.

/2/ what forms the real basis of algebraic linguistics:

N. Chomsky's merit is, first of all, that he establishes a relation between the generation or recognition of the language /as a set of sentences/ and the structural descriptions provided for the sentences. Thus, it becomes possible to describe natural languages with the help of the notions and methods of algebraic and mathematical logic and thereby to compare the different types of grammars with the different types of automata by means of mathematical methods.

/3/ the application of the approach to natural languages by the method described in point /2/:

Transformational grammar is the most developed system for the description of language in a generative way and has many advantages over other similar systems. In spite of this, the mathematical formulation, the linguistic interpretation and the practical application of transformational grammar raises many difficulties. The cause of this, on the one hand, is that there is no formal definition of TG taken as a whole, and on the other, in linguistic interpretation the problem of the levels of the linguistic system emerges as well as the relation of grammar to semantics. While in the application in practice the unsolved problem of the identifying algorithm applicable in the case of the transformational grammar causes some difficulties. Therefore, it would be desirable to work out some other alternative for the generative description of language. That is why the possibility to approach the description of language with the aid of a system consisting of a sequence of pushdown-store transducers is being examined.

/4/ the interrelation of algebraic linguistics and its application:

It cannot be said that the algebraic theory of grammar owes its existence to its direct relation to machine translation, but it is undoubtedly important from the viewpoint of machine translation and from that of some similar application as well. These kinds of application give us the possibility to try to describe the different languages in an algebraic way, i.e. the mathematical approach to semantics could hardly be realized without the mass processing of different texts with the use of computers.

The article written by B. Palek, P. Piťha and P. Sgall, and entitled the "Mathematical Linguistics in Czechoslovakia" / 1 /1964/, pp. 9-23 / gives us a general idea of the situation of mathematical linguistics in Czechoslovakia. Though this article reflects the 1964 situation we think it would be useful to give a short outline of its contents. V. Mathesius carried out statistical and phonological reasearch works, he has compared the frequency of the Czech phonemes with that of other languages. He has also dealt with the historical development of word order in English.

As it is known, B. Trnka also wrote a work about the frequency of the English phonemes. The following scientists have also carried out certain examinations in connection with phonology: J. Krámský, J. Vachek, V. Mázlová,

A. A. Isačenko, V. Skalička. They have applied statistical methods not only in phonology but also when examinig some other kinds of linguistic characteristics. From among these the following researchers should be mentioned: V. Šmilauer, P. Poucha, V. Fried.

The first international bibliography on statistical linguistics was published by B. Trnka.

In the Department of Czech Language, Faculty of Philosophy, Charles University, Prague, it was the Algebraic Linguistics and Machine Translation Department which in 1958 began to carry out English-Czech machine translation activities. In the analysis, among others, they have solved the problem of the homonymy of the -ed verb form, while in the Czech synthesis they have succeeded in solving the problem of the adjective-noun and subject-predicate agreements.

In 1961 The Linguistic Group of the Centre of Numerical Mathematics joined the Faculty of Mathematics and Physics at Charles University. In co-operation with the Computing Centre of the Economics Institute they have worked out a process for establishing the frequency of verb forms and they have compiled a minor frequency dictionary for mathematical texts.

They have also dealt with the algorithm of the synthesis and analysis of the Czech language /independently of the target language/. They have tested the synthesis of the declension of Czech nouns on an LGP-30 type computer.

In 1961 a mathematical and applied linguistic department was formed within the Institute of the Czech Language of the Czechoslovak Academy of Sciences. Further, in 1962, a Mathematical Linguistic Department was established at the Institute of Slovak Language of Slovak Academy of Sciences, the task of which is to deal with the statistical and structural examination of the Slovak language.

Mathematical linguistic problems are being dealt with also at the Linguistic Department of the Czechoslovak-Soviet

Institute of the Czechoslovak Academy of Sciences as well as at the Department of Grammar and Stylistics at the Institute of the Czech Language, where in the course of linguistic research works punched card machines will be applied.

Ladislav Nebeský is his article "On a formal grammar" published a formal grammar / 1 /1964/, pp. 24-28 / which differs essentially from Chomsky's phrasestructure grammar. This system is suitable for constructing certain simpler types of languages.

The article of L. Nebeský and P. Sgall entitled "The relation of "form" and "function" in language /summary/ / 1 /1964/, pp. 29-39 / gives us a short summary of their work having the same title and published in the periodical "Slovo a slovesnost" / 23 /1962/, pp. 174-189 /.
In this work of theirs they establish an axiom-system /in order to be capable to examine linguistic structure with more exact means than it used to be done by traditional linguistics up till that time, but at the same time in such a way that the defined notions should cover the notions of traditional linguistics as exactly as possible/ which defines the notions "form" and "function". These notions express the existing relations of the different levels of the linguistic system to each other. By the help of this axiom-system certain linguistic notions can be defined /homonymy, synonymy/ as well as the hierarchy of the linguistic elements, and also the notion of duality of the linguistic system. The notions "form" and "function" are suitable to elucidate some linguistic relations.

Ladislav Nebeský in his article entitled "A sentence analysing model" / 2 /1964/, pp. 3-10/ writes about a model of sentence analysing. In one of his former articles he defined this model as follows: there is a set of sentences and a set of rules. With the aid of the rules we can add a set of sub-sentences to each sentence /we get the sub--sentences from the original sentence by leaving out one word form at a time of the original sentence when forming each of the sub-sentences/. By this method we can define the linear relations within the sentence, the task of which is the modelling of sentence analysis /of the dependency graph of the sentence/.

This article ofters a general and formal formulation of the relations of subordination illustrating the thesis with a concrete Czech example.

Pavel Novák in his article "Two types of formulae in quantitative linguistics" / 2 /1964/, pp. 11-14 / deals with the two main problems appearing in the course of the application of the quantitative apparatus, such as /1/ the adding of numbers /quantitative indices/ to experimental facts, and/2/ the establishment of the relations among the quantitative indices referring to the experimental facts.

O. Leška and A. Kurimskij in their article entitled "Entropy in the language" / 2 /1965/, pp. 15-21 / after enumerating the tasks of theoretical, general, special, experimental and applied linguistics, state that the entropy of language is a manysided problem and its examination belongs to a certain degree to each of the above mentioned branches of science. When examining the problem of entropy we have to start out from the followings: /a/ of the relation of the denotatum and designatum, /b/ of the socalled functional--constitutive rules /these determine the selection of the single allomorphemes when forming a word form and /c/ of the so-called functional rules of linguistic stylization /these determine the classification of the invariants of the language system according to functions/. The examination of entropy might cover also several different aspects of the language, and the results are significant not only from the viewpoint of the special but also from that of general linguistics as well.

J. Jelínek and L. Nebeský published an article on "A syntactic analyser of English" / 2 /1965/, pp. 22-33, -

3 /1965/, pp. 38-59, - 4 /1965/, pp. 62-69, - 5 /1966/ pp. 31-61/. The article goes into much detail, and it is continued through several volumes. Such a detailed description of the examination methods, the explanation of the block-systems, the enumerating of lists as can be found in this article is perhaps not expedient because researchers in general do not need a detailed study concerning the practical solution of a single problem. The review describes the syntactical analysis of English, which problem is being dealt with at the algebraic linguistic and machine translation department of the faculty of linguistics and phonetics of Charles University in Prague. It discusses mainly methodological aspects.

The sections of the article:

1. preliminary data
2. description of the block system
3. elaboration of the block system
4. interpretation of the block system
5. some possibilities of the technical realization
6. the outline of the operation of the algorithm

The author discusses in detail the block system of the English nominal structures, the word class list belonging to it, and the list of the most important blocks.

Dana Konečna in her article entitled "The analysis of Czech verb forms" / 2 /1965/, pp. 34-51 / used a morphological analysis according to which the word to be analysed is to be divided into two parts: the stem and the ending, and the ending should be regarded as the starting point of the analysis. In the case of the Czech verbs the last phoneme of the ending renders us enough information of grammatical character that the analysis should give the right information. The algorithm of the analysis described in the article was devised for an LGP-30 type computer. When feeding the verbs into the computer we also give information concerning: 1/ the word form, 2/ the different verb classes, 3/ the aspect of verbs, 4/ the length of the ending. The last phoneme of the ending shows us at the same time that where can we find the necessary instructions for the analysis of the forms ending in the respective phoneme. In the case of compound verb forms analysis begins with the analysis of the constituents. When the analysis of the constituents on the basis of the last phoneme is finished the meaning of the compound verb form as a whole can be established with the application of certain rules. This analyzing method is regarded by the author as significant from two points of view: on the one hand, it brings to the surface some typological characteristics of the Czech verbs, and on the other, it clearly shows some specific features of the semantic structure of the compound verb forms.

Ladislav Nebeský in his article entitled the "Conditional replacement of words" / 3 /1965/, pp. 3-12/ deals only with the relations between the words /and not between the words and word sequences/. In order to have the mathematical definitions and formulae understood we should have to make a more detailed review on the article. The article, for example, does not make it quite clear that to what kind of set do the complementary sets refer. That is why we omit a more detailed discussion of the theory expounded in the article.

The article of N. A. Paščenko, entitled "The analysis of expressions with an adverb of time semantic contents in the Russian and Czech languages." / 3 /1965/, pp. 13-37, 4 /1965/, pp. 26-61/ deals with the analysis of noun /partly adverbial/ structures denoting adverb of time with or without preposition. The structures of this kind present a lot of difficulties in the course of machine translation because it is not at all easy to process them with formal methods because of their individual character. The article contains the systematization of the Czech and the equivalent Russian expressions of temporal meaning and also the rules of equation. To every meaning more than one syntactic structures corresponds, each of which is lexically determined. To the class of the grammatical structures /or simply structures/ certain deifferentiating semantic marks, characteristic to the given semantic contents, are attached. The class of the structures can be so considered as if they were a set of synonymous structures which consists of the sub-sets of noun structures with or without preposition and also of subsets of adverbs not intersecting each other. Of these three subsets any one may be empty. The author differentiates nine groups denoting time relations. The syntactic way of expressing the time meaning depends on the semantic contents of the words included in the structure. Consequently the nouns used in the time adverbial expressions can be divided into two great semantic classes: a/ the class of those nouns which already in their form express a point or duration of time /hour, winter, February/, and b/ the class of those nouns which in directly but nevertheless also express time through an activity, phenomenon or situation having a certain temporal implication /work, lunch, revolution/. It is evident that the polysemy of the words should disappear only in the course of the analysis of the immediale context. The two great semantic groups can be divided into further sub-groups. The article later gives a detailed analysis of the categories expressing time. The appendix contains the list of the differentiating semantic features, the classes of the structures and the rules of equating the Czech structures with the Russian ones.

Jarmila Panevová in her article on "The analysis of electro-technical texts", / 4 /1965/, pp. 3-25 / gives an account of the results of the linguistic analysis of technical texts. With the consideration of several viewpoints eighty-two pages of electro-technical text in Czech language were encoded on punched cards. In selecting the material they have taken into consideration the demands of three concrete tasks: the preparation of the sythesis of Czech language for the purposes of machine translation /1/, the composition of the intermediary language /2/, the generative description of the Czech language /3/.

The first part of the article contains the description of the system of analysis while the second analyzes the results and problems of the analysis. The units of the analysis are the single words together with the auxiliary words belonging to them. Every unit was encoded on punched cards. The analysis of the sentences is carried out on the basis of principles of dependency grammar, the only independent constituent being the predicate of the main sentence. With the help of the data the following can be defined: /1/ the function of the words from the viewpoint of the semantic structure of the sentence, the subject and the object of the action, the different determinants and the secondary predicates, /2/ the single classes of words, /3/ the morphological categories of the words. Furthermore the article includes some data referring to the government and to the surroundings of the word. The classification of the material is carried out on four levels. We shall not here analyse the results.

Antonin Řiha in his article "On the recognition procedure for pushdown store transducers" / 5 /1966/, pp. 3-15 / takes as a basis the pushdown store machine described by R. J. Evey in his work "The Theory and Application of Pushdown Store Machines" /Report No. N55-10, The Computation Laboratory of Harvard University, Cambridge, Massachusetts, 1963/. Therefore, for the sake of better understanding, he cites some definitions and theses of the above-mentioned work. Then, as a thesis, he renders a pushdown store transducer P in such a way that a recognition system should be its inverse. For this he proves that for a given $\varphi \in L$ sequence the maximal number C /as the function of the sequence/ of those steps which are important to decide whether $\varphi \in f/L/$ is, or not.

If the condition of the thesis is fulfilled on a certain symbol, then, he states, the P transducer keeps the respective symbol, if not, the symbol disappears. Making certain conditions through the use of these notions, binding on the feed-in sequence of the P transducer, it can be achieved that the P transducer should become a recognition system. This is proved by him as a separate thesis. The two theses provide a satisfactory condition for making an inverse pushdown store machine suitable for the recognition process.

In an other thesis he gives the necessary and satisfactory condition for a pushdown store machine to become deterministic, namely, the deterministic inverse transducer can be rather well used for the recognition process. Here the conditions of the thesis refer to the form of the rules of the pushdown store machine.

A. Liudskanov and E. Paskaleva in their article "The lexical problems of the machine translation of a Russian language mathematical text into Bulgarian", / 5 /1966/, pp. 16-30, 6 /1966/, pp. 27-34 / make an account of the problems of the dictionary composed by the Mathematical and Computation Technical Centre of the Academy of Sciences of Bulgaria. The aim of the article is to describe the general principles of composing a dictionary /1/, to raise and solve the emerging problems /2/, the short statistical and structural description of the dictionary /3/.

Petr Sgall gives the resume of his work "Generativni popis jazyka a česka deklinace" published in Czech, in an article in volume 6 of the Prague Bulletin, under the title "Generative description of language and the Czech declension" / 6 /1966/, pp. 3-18 /. The aim of the book is to give a new alternative for the generative description of the natural language which basically agrees with Chomsky's generative grammar but does not apply any transformation rules. The author discusses semantics as an essential component of the "grammatical" description and not as a separate component added to the grammar. With this solution he wants to avoid the difficulty which is posed by the differentiation between the syntactic structure itself and its own semantic representation.

The second part discusses the linguistic conditions of the descriptions of the language on several different levels. The starting point of the author is the hypothesis that there are at least two levels of sentence structure. The first is the so-called semantic sentence structure /with syntactic relations/. On this level, which he calls a tectogrammatical level, there are besides the units corresponding to the syntactic relations also lexical and morphological units. The other level of sentence structure is the phonogrammatical level. The single constituents of this are also divided into three types: the syntactical relations and the lexical units have a different character on this level from those on the tectogrammatical level, while the morphological units /co-ordination, apposition, ending/ and the corresponding units of the tectogrammatical level can be mutually and unambiguously made to correspond to each other. The third level is the morpheme level, the units of which are the stem-morphemes, the cases and the prepositions, etc. Presumably there are even further levels to be differentiated. Undoubtedly we can speak of a morphophonemic and a phonemic level as well.

The third part discusses the main formal characteristics. The system has several components. The first is productive /generative, recursive/, the others are transductive. The generative component is a context free sentence structure grammar. The transductive part consists of further four constituents. These keep the dominant symbol until symbols depending get on the outcoming tape. These four pushdown store transducers will be followed by two or three finite state transducers which keep on transducing the representations of the sentences from one level to the other till they reach the phoneme level.

The fourth part attempts the description of Czech noun declension with the help of forty-two morpheme and twenty-one phoneme rules and at the same time taking into consideration the requirements of machine translation.

Pavel Novák has written an article entitled "On a model of the stylistic component of language encoding" / 6 /1966/, pp. 19-26 / based on L. Doležel's work "A model of the Stylistic Component of Language Encoding", published in 1965. / L. Doležel has published a whole series of articles on language encoding in which he discusses mainly the stylistic component. This component explains the so-called style-characteristics./. Here the author deals, first of all, with that formal system which serves as a model for the operation of a selector. Then he examines different inter relations. For example those existing between the stylistic component and the stylistic characteristics, and between the stylistic characteristics, their peculiarities and the stylistic component of language encoding.

Ladislav Nebeský has contributed an article with the title "On the notion of relevant features", / 6 /1966/, pp. 35-44 /. In modern linguistics the theory of linguistic levels becomes more and more important. On this level one of the most decisive problems is the formation of more general linguistic units out of more concrete units. Such a system has been worked out in a more detailed way on phonological level. /S. Marcus, 1963, Un Model matematic al fonemului, St. Cercer.matematice, 14, 3, 405-421 /, /S. Marcus, 1963, Linguistica matematică, București, 56-68/. In connection with this the author is dealing with the model of Marcus. He wants to point out that the notion of the relevant characteristics of the lingusitic units are not clear enough, and it is very difficult to find a solution satisfactory from every point of view.

Finally, we may read a review by Dušan Pospíšil, entitled "On a linearization of projective W-trees", / 6 /1966/ pp. 44-68 /. First of all, the author discusses some primary notions of the graph theory because he considers the use of graphs to be useful when describing sentence structures. After this, he defines the projective W-tree having marked peaks, then the L_V language. He proves the thesis that the representation in the L_V language is unambiguous.

Summing up our impressions gained while studying the different treatises, while we emphasize the value of the volumes, we would like to point out some insufficiencies. As we have already mentioned the publication of the same article in instalments makes understanding difficult. Printing errors in more than one article are also disturbing. The same refers to the lack of explanation of certain symbols and marks. In spite of these defects and as illustrated by the reviewed articles, the researches, and their results accomplished by the scholars of the Prague Charles University and other research institutes in the field of mathematical linguistics, have furthered this young branch of science in its development to a very great extent.