

MS 19. 717

AP63

12397 R

23-1500  
-11

# COMPUTATIONAL LINGUISTICS

# I

COMPUTING CENTRE OF THE HUNGARIAN ACADEMY OF SCIENCES  
BUDAPEST, 1963



COMPUTATIONAL LINGUISTICS

I

COMPUTING CENTRE OF THE HUNGARIAN ACADEMY OF SCIENCES

BUDAPEST, 1963

Szerkesztette: Kiefer Ferenc

Felelős kiadó: Frey Tamás

---

Alak: A/4      Ivszám: 33,5 - -

Megjelent 1963. június hóban, 500 példányban

---

Sokszorosította a Főv.Nyomdaipari V. 16. telepe - 1239

## P R E F A C E

In Hungary the actual work of programming for machine translation was started in 1961 at the Chair of Foreign Languages at the University of Technical Sciences, Budapest. However, there was no computer of a capacity suitable for holding an all-embracing program at the disposal of the University. A larger computer, the personal and financial support given by the Computing Centre of the Hungarian Academy of Sciences made this institute a new and suitable centre of the pursuance of the work.

The most important aim of the research work carried out in connection with the problems of machine translation is to make up and try an algorithm for translating from Russian into Hungarian, but, besides the theoretical research work necessary for the experiments, the general problems of mathematical linguistics belong to the labour plan of the group of the research workers as well.

An account of the work performed is to be given once or twice a year in special publications by the Computing Centre. The first issue deals with some questions of mathematical linguistics and makes known the initial results achieved in making up an all-out algorithm for translating from Russian into Hungarian: the analysis of nominal constructions containing several members and that of the grammatically homonym nominal forms. The articles deal with the possibilities of how the problems mentioned above can be solved by a computer without reference to any practical results as yet.



ОПРЕДЕЛЕНИЕ НОМИНАЛЬНЫХ ГРУПП  
В МП С РУССКОГО НА ВЕНГЕРСКИЙ

Г. Хэлл

І. ОБЩИЕ ВОПРОСЫ

1. МП, кибернетика и лингвистика

Изучение отношения МП к кибернетике и лингвистике представляется нужным в первую очередь с точки зрения МП. Правда, МП является еще молодой отраслью науки, но тот факт, что статьи о вопросах МП появляются в журналах с различным профилем, довольно однозначно говорит о неопределенности того, куда относить М П.

Осуществить человеческую работу перевода с одного языка на другой электронной /вычислительной/ машиной - это основная задача М.П. Дисциплина, которая поставила целью изучать характерную деятельность живых организмов, в том числе человека и стремится подражать этой деятельности электронными машинами, - кибернетика. В основе кибернетики лежит тот факт, что в различной деятельности живого организма существенную роль играет циркуляция информации и их преобразование при этом. Этот процесс может быть математически описан алгоритмами. Самым волнующим и при том в его перспективах самым значительным вопросом кибернетики является изучение проблемы, какие возможности имеют электронные вычислительные машины в подражании мышлению человека. Эта задача повидимому не может быть решена простым имитированием на машине деятельности человеческого мышления во всей его сложности. Более надежным кажется метод, по которому следует механически приблизить в первую очередь те специальные области мышления, которые уже на первый взгляд легче описать формулами алгоритмов. К таким областям мышления относятся прежде всего умственная деятельность, связанная с вычислением, некоторой игрой, логическим умозаключением и переводом с одного языка на другой [1].

Таким образом мы считаем, что МП является отраслью кибернетики. Его первичная задача - описать в алгоритмической форме характерные умственные деятельности человека при переводе. От этой задачи конечно нельзя отделять изучение языкового материала. М.П. должен ставить вопрос: Что такой язык, каков язык, правильнее, каковы те языки, с которых или на которые осуществляется перевод? При этом М.П. ставит эти вопросы с узко ограниченной точки зрения: М.П. хочет знать, как можно описать грамматический разбор языкового текста и правила перевода в формах алгоритма.

Имея в виду труды по лингвистике и, главным образом, по вопросам перевода, можно сказать, что М.П. берется за механическое осуществление такой человеческой деятельности, которая до сих пор в сущности мало изучена. Значит, речи о том, как превращать уже хорошо известную деятельность перевода в механическую и подражать работе человеческого мозга с помощью машины, быть не может.

В случае М.П. кибернетика поступает иным путем: прикладное естествознание создало машину, которая может работать "умно", т.е. выводить результаты не только складыванием, вычитанием, умножением и делением, а так же способна осуществить так называемые логические операции, и с помощью них может взвешивать, как бы умно обдумывать, подобно человеку. М.П. может пользоваться этими возможностями, но только этими.

Сопоставляя специальные области исследования языкознания и М.П. можем сказать, что:

1/ первичным предметом лингвистики является языковой материал, а М.П. интересуется им только второстепенно, поскольку умственные операции при переводе не могут быть отделены от языка;

2/ языкознание не ограничивается при своих исследованиях такими узкими принципами как М.П.



## 2. Основные вопросы методики М.П.

В связи с методом М.П. с русского на венгерский /в дальнейшем М.П. р-в/ кажется целесообразным присоединиться к высказываниям, находящимся в работах по методам М.П.

Различают 95 %-ный и 100 %-ный подходы [2]. Первый метод применяется в сущности при пословном переводе, когда машина производит только очень ограниченный и относительный разбор, нужный для составления перевода в используемой форме. Так как М.П. с таким методом отодвигает грамматический разбор на второй план, применять такой подход можно только между языками, структура которых имеет большое сходство, или если в выходном языке порядок слов имеет не очень важное значение с точки зрения правильности перевода.

В М.П. между языками, имеющими большое расхождение в порядке слов, напр. в русско-венгерском переводе, пословный метод не может быть применен даже в начальной стадии работы.

100 %-ный метод осуществляет перевод даже простого переводимого текста на основе подробного грамматического анализа. По этому методу грамматический анализ является центральным вопросом перевода, а не просто средством улучшения "переведенного" текста.

М.П. р-в придерживается второго метода, т.е. составления алгоритмов на основе подробного и распространенного грамматического разбора, не имея в виду при этом никакого конкретного материала, только характерные черты русских научных и технических текстов.

По мнению многих по существу нельзя провести редкой границы между 95 %-ным и 100 %-ным методами, ведь с помощью общепринятых грамматических категорий и теорий языка не удастся найти процедуру анализа, правильную и действительную

для всех фактов языка, даже если факты имеют довольно общую интерпретацию. А если не обращать внимания на то, что метод т.н. 95 %-ного перевода использует грамматический разбор для улучшения, а не для обоснования перевода, то разницы между двумя методами действительно исчезают, так как методы встретятся где-нибудь в степени и распространенности разбора.

Согласно алгоритму М.П. Университета в Вашингтоне /алгоритм может быть принят образцовым алгоритмом дословного перевода/ составлению лексикона и решению лексических вопросов уступает в важности грамматический анализ [3]. Подробным лексиконом, если в его состав входят не только основные, но и все релевантные формы, можно решать и вопросы синтаксиса. Проблемы, возникающие при этом в связи с очень большим словарем, могут быть решены с помощью т.н. фотоскорической памяти. Такой подход к решению задачи М.П. уже значительно различает 95 %-ный метод от 100 %-ного, который ожидает получение правильного перевода от подробного грамматического анализа. 95 %-ный метод, принимая во внимание его результаты, не заслуживает того недооценивания, которое имеется в его названии.

При подготовке М.П. с русского на венгерский лексические вопросы к сожалению не разработались.

### 3. Вопросы языка-посредника

Программа М.П. с русского на венгерский предусматривает непосредственный перевод между двумя языками, без применения т.н. языка-посредника.

Не входя в подробный разбор вопроса языка-посредника, мы считаем нужным говорить коротко о тех соображениях, на основе которых при подготовке М.П. р-в отказались от при-

менения языка-посредника. /Термин "язык-посредник" применяется здесь с тем содержанием термина "интерлингва", как это предложил Н. Д. Андреев [4]. /

В пользу применения языка-посредника говорит то безусловно правильное рассуждение, что число нужных алгоритмов сильно уменьшается. Против применения говорит чрезвычайная сложность анализа. Несомненно, разработка алгоритма, дающего перевод в применяемой форме, является более реальной и легче осуществимой целью, чем решение сложной проблемы языка посредника. Кроме этого успешное решение непосредственного перевода с одного языка на другой является как бы предпосылкой возможности создания языка-посредника. Осуществление такого промежуточного машинного языка невозможно до разработки подходящих программ М.П. между двумя языками.

Если сопоставить общую схему М.П. р-в со схемой перевода с применением языка-посредника, то по Н.Д. Андрееву:

$$A \longrightarrow \text{МЯ} \longrightarrow \text{ЯП} \longrightarrow \text{МЯ} \longrightarrow B$$

A = входной язык  
B = выходной язык  
МЯ = метаязык  
ЯП = язык посредник

Метаязыки дают морфемы, грамматические и семантические связи этих морфем для обоих языков в полном богатстве. Язык-посредник - интерлингва - содержит эти информации только в какой-то обобщенной форме.

Схема М.П. р-в :

$$A \longrightarrow \text{"МЯ"} \longrightarrow B$$

В "метаязыке" в М.П. р-в не содержится полный разбор русского языка. Анализ делается только при соблюдении потребностей синтеза венгерского языка.

## II. ВОПРОСЫ АЛГОРИТМА МП Р - В

### 1. Семантические и грамматические основы

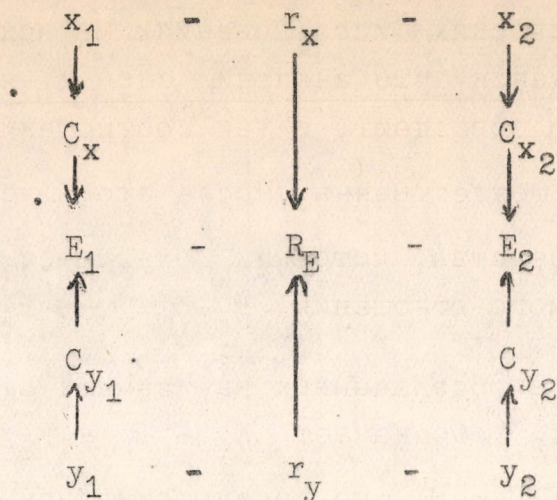
#### МП р - в

К переводам, особенно к переводам технических и научных текстов, предъявляют требования, чтобы содержание перевода совпадало с содержанием исходного текста. В то же время лингвистические работы утверждают, что при переводе нельзя достигнуть полного соответствия. Причины этого кроются в семантических и грамматических расхождениях между языками, вследствие чего разными языками трудно выражать тождественные вне языка факты и соотношения. Считаться с этим приходится в каждом переводе.

Работы по переводу утверждают и то, что наиболее точные переводы возможны в области научной и технической литературы. Семантические свойства словарного фонда сходятся у таких текстов больше, чем у текстов с общеупотребительной лексикой, а грамматические средства, хотя и сложные, все-таки имеют более однородный характер, чем у литературного и разговорного языка.

В разработке программы МП мы еще не могли заниматься расхождениями семантического характера между русским и венгерским языками. Нашей первичной целью было выяснить соотношения в грамматических средствах, служащих выражения идентичных реальных /внеязычных/ сношений.

Сказанное может быть представлено графиком [5], изображающим семантические и грамматические понятия, а также обозначенные объекты действительности:



$x_1$  и  $x_2$  - слова языка

$r_x$  - синтагматическая связь между словами  $X_1$  и  $X_2$ ;

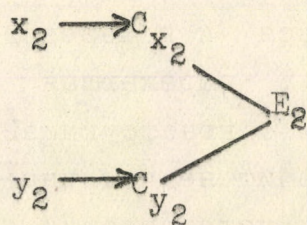
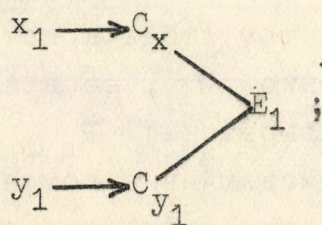
$y_1 - r_y - y_2$  - синтагма в переводе, при том члены соответствуют по значению  $X_1 - r_x - X_2$ ;

$E_1$  и  $E_2$  - предметы реального мира или элементы внешней действительности;

$R_E$  - реальное соотношение.

$C_{x_1}$  и  $C_{x_2}$ , а также  $C_{y_1}$  и  $C_{y_2}$  дают значения слов

$x_1, x_2, y_1$  и  $y_2$ .



дают

слова и их полные значения, т.е. кроме графической /звуковой/ формы слов и значений еще обозначенные предметы.

При грамматических исследованиях мы исходили из того упрощенного положения, что значения слов  $x_1$  и  $y_1$ , а также  $x_2$  и  $y_2$  полностью совпадают, и так соотношения  $x_1 \rightarrow C_{x_1} \rightarrow E_1$  и  $y_1 \rightarrow C_{y_1} \rightarrow E_2$  тождественны. После этого установили те грамматические средства, которыми пользуются языки при выражении того же самого отношения  $R_E / r_x \rightarrow R_E \rightarrow r_y /$ .

Расхождения в обозначенных внеязычных отношениях /т.е. когда  $x_1 - r_x - x_2$  обозначает  $E_1 - R_E - E_2$ , а  $y_1 - r_y - y_2$  обозначает  $E_1 - R'_E - E_2$ / не рассматривались, если они в венгерском переводе не дали существенного расхождения. /Напр., соотношения, выраженные определенным артиклем, объективным спряжением и т.д./

Грамматические соотношения между словами были введены к связям между классами слов, чтобы получить систему, удобную для МП. Грамматические связи между единичными словами имелись в виду только при исключениях.

Классификация русского и венгерского словарного состава, установление подклассов и групп, даже грамматических свойств одних слов осуществлялось с такой целью, чтобы классификация и грамматические особенности служили основой к формальному переводу. Последнее требует установления различных типов перевода на основе соотношения конструкций  $x - r_x - x_{i+1}$  и  $y_i - r_y - y_{i+1}$ , из которых каждая пара выражает одно  $R_E$ , а также установления определенных конструкций  $x - r_x - x$ , отождествляемых формальным путем и выражающих все те  $R_i$ , которые должны выражаться и в переводе. При том любая конструкция  $x_i - r_x - x_{i+1}$ , выражающая отношение  $R_i$ , не должна по форме совпадать с конструкцией для выражения  $R_n$ . Другими словами это значит не что иное, как исключение гомонимных конструкций при установлении синтагматических отношений.

С точки зрения перевода конструкции  $x - r_x - x$ ;  $y - r_y - y$  для выражения тождественной  $R$  могут быть разделены в два типа [6]:

- 1/ равные конструкции,
- 2/ трансформируемые конструкции.

Равными называются конструкции, в которых морфемы, выражающие не грамматические отношения, входят в соответственно равные классы слов в обеих языках. Расхождение в порядке слов не имеет значения при определении типа конструкций.

Равные конструкции, напр:

$$A + S \longrightarrow A + S$$

$$V + S \longrightarrow V + S$$

$$S + S_a \longrightarrow S_{/a/} + S_g$$

$$P_{r_x} + S_x \longrightarrow S_y$$

$$P_{r_x} + S_x \longrightarrow S_{/y/} + P_g$$

и т.д.

---

Обозначения:	A	=	прилагательное;
	S	=	существительное;
	S <sub>a</sub>	=	существ. в родительном падеже /или с суффиксом, обозначающим обладателя/;
	S <sub>g</sub>	=	существ. с суффиксом, обозначающим обладаемое;
	S <sub>x</sub>	=	существ. в косвенном падеже или с суффиксом;
	V	=	глагол в личной /родовой/ форме;
	P <sub>r<sub>x</sub></sub>	=	предлог с управлением x;
	P <sub>g</sub>	=	послелог;
	/ /	=	форма может отсутствовать.

---

Трансформационными называются конструкции, в которых морфемы, выражающие не грамматические соотношения, входят в разные классы слов. Таковы напр.:

$$V + V_{inf} + S \longrightarrow V + C + R + V + S$$

$$S_x + C + P_x P_{ry} + S_y + C \longrightarrow S_x + C + K + V + S_z + C$$

и т.д.

---

Новые обозначения:

- С = запятая,  
 К = союз,  
 Р = причастие.
- 

Применяемые формальные методы грамматического анализа часто наступают на гомонимные конструкции. Поэтому стремление установить для каждого отношения  $R_i$  в языке-источнике отдельную конструкцию  $x_i - r_x - x_{i+n}$ , оказывается очень важным требованием. До какой степени нам это удастся, еще не известно. Работа ведется в направлении установления соответствующих классов и подклассов, учитывания порядка слов и, может быть, семантического значения. Несомненно, устранение гомонимности при формальном анализе очень облегчило бы не только анализ текста, но и весь ход МП.

Анализ исходного текста, на основе выше сказанного, состоит из разбора, по которому устанавливается класс /морфологический класс/ слова, его конкретная грамматическая /парадигматическая/ форма. Из этих данных можно получить информации по функции слов. Конечно, мало таких словоформ, которые сами по себе дают достаточные данные к "равному" или "трансформационному" переводу. В остальном большинстве случаев требуется выяснить грамматическую связь слова с другими, окружающими его словами.



В обнаружении синтагматических соотношений  $x - r_x - x$  с точки зрения перевода на венгерский язык особенно важно установление того, какую связь /прямую или косвенную/ имеет слово к слову, могущему представить предложение в целом, т.е. сказуемому предложения по терминологии традиционной грамматики. Есть такие слова, у которых уже по классу /деепричастия, некоторые модальные слова и союзы/ или по парадигматической форме /существительное женского рода в именительном падеже/ можно определить это отношение, но для большинства слов требуется выяснить и конструктивные связи.

## 2. Кодовая система

Первые операции МП р - в :

- 1/ Поиск слова,
- 2/ установление класса слова,
- 3/ установление парадигматической формы слова.

Поиск слов не представляет особую операцию в МП р - в. Из практики других групп по МП для нас казался самым приемлемым метод Этингера [7] для определения окончаний; мы добавили к этому методу лишь анализ слов с правописанием через тире. Для анализа слов, у которых последние буквы /основы/ похожи на окончания /напр. не, но, для, предмет, натрий и т.д./, применяется два способа: по первому неполные основы могут входить в словарь основ (напр. дл/-я/, д/-о/, п/-о/, н/-о/, н/-е/ предм/-ет/ и т.д.) , по второму способу из этих слов можно образовать особую группу слов, группу т.н. "полных слов", и каждое слово, имеющее соответствующую длину, должно согласовываться составом этого словаря.

Слова, имеющие тире по правописанию, автоматически разделяются у тире и после удаления последнего, слово подвергается анализу на окончание /подобно анализу слов с окончанием на -сь или -ся/. Венгерские эквиваленты этих слов получают особый индекс.

Информации о словах фиксируются в машине в форме кодовых чисел. Так как при выборе системы кодовых чисел целесообразно иметь в виду технические возможности машины, предназначенной для перевода, то кодовая система МП р - в - и вследствие этого в какой-то мере и грамматический анализ зависит от свойств машины М-3, применяемой при исследованиях.

В программировании грамматических информаций обычно применяется два способа:

- а/ каждой информации приписывается один определенный разряд,
- б/ различные информаций получают различные кодовые числа.

Из экономических соображений мы применяли вторую возможность. Так в виду технических свойств машины М-3 каждая группа информаций может содержать в крайнем случае 63 различных информаций.

Классы русских слов и кодовые цифры для МП р - в

Обозначение информации: К. I.

Класс существительных:

Грамматическое обозначение	Кодовые цифры	Информация			
		1	2	3	
S1	1	мужской	род на тв.согл	-й, -ь	неодушевл.
S2	2	"	" " " "	" "	одушевл.
S3	3	"	" на-ий		неодушевл.
-	4	-	- - -		-
S5	5	женский	род на-а, -я		неодушевл.
-	6	-	- - -		-
S7	7	"	на -ия		неодушевл.
S8	8	"	на -ь		"
S9	9	средний	род -о, -е		"
S10	10	"	на -ие		"

1	2	3
S11	11	мужской род множ.ч. Им.п. на-а неодушевл.
S12	12	средний р. " " " -ья "
S13	13	мужской род.ед.ч. предл.п.на-у "
S14	14	" множ.ч. им.пд. на-ья "
S15	15	" " " " " " одушевл.
S16	16	"путь"
S17	17	"время"
-	18-27	

Класс глаголов:

Грамм. обозн.	Код.цифры	И н ф о р м а ц и я
V1 - V18	28 - 43	полнозначные глаголы /основы наст. и прош.вр. отдельно/
Vau	44	вспомогательные глаголы-основы наст. вр. и прош. вр.
V19 - V21	45 - 47	полнозначные глаголы

Класс причастий:

Грамм. обозн.	Кодовые цифры	И н ф о р м а ц и я
P5	48	страд.прич. прош.вр. в краткой форме
P1	53	действ. прич. наст.вр.
P2	54	действ. прич. прош.вр.
P3	55	страд. прич. наст.вр.
P4	56	страд. прич. про. ср.

Класс прилагательных, качественных указ. местоимений и  
порядковых числительных:

Грамм. обозн.	Кодовые цифры	И н ф о р м а ц и я
A1	49	окончание на -ой, -ая, -ое
A2	50	" на -ый, -ая, -ое и -ий, -ая, -ое
A3	51	" на -ий, -ая, -ее и -ий, -ая, -ее
A4	52	" на -ий, -ья, -ье

Класс местоимений.:

Грамм. обозн.	Кодовые цифры	И н ф о р м а ц и я
Pn1	57	Личные местоимения
Pn2	58	Относительное местоим. /который, -ая, -ое/
Pn3	59	Притяжательные местоим.
Pn4	60	Местоимения "этот", "тот"

Класс числительных:

Грамм. обозн.	Кодовые цифры	И н ф о р м а ц и я
Nu	61	"один" + числ. опр. и неопр.

Класс различных модальных слов:

Грамм. обозн.	Кодовые цифры	И н ф о р м а ц и я
PD	62	Модальные слова

Смешанный класс

Грамм. обозн.	Кодовые цифры	И н ф о р м а ц и я
MS	63	

Парадигматические формы русских слов

Обозначение информации: К. II.

Парадигматические формы существительных: К I = I-27

Грамм. обозн.	Кодовые цифры	И н ф о р м а ц и я	
msN	1	мужской род ед.ч.	именит. над.
msA	2	"	винительный п.
msG	3	"	родительный п.
msD	4	"	дательный п.
msI	5	"	творительный п.
msP	6	"	предложный п.
fsN	7	женский р. ед.ч.	именительный п.
fsA	8	"	винительный п.
fsG	9	"	родительный п.
fsD	10	"	дательный п.
fsI	11	"	творительный п.
fsP	12	"	предложный п.
nsN	13	средний р. ед.ч.	именительный п.
nsA	14	"	винительный п.
nsD	15	"	родительный п.
nsD	16	"	дательный п.
nsI	17	"	творительный п.
nsP	18	"	предложный п.
plN	19	множеств. ч.	именительный п.
plA	20	"	вин. п.
plG	21	"	род. п.
plD	22	"	дат. п.
plI	23	"	творит. п.
plP	24	"	предложный п.
smNA	25	ед.ч.м.р.	им.+ вин. пп.
smDP	26	" "	дат. + предл. пп.
smAG	27	" "	вин. + род. пп.
sfDP	28	" женск.р.	дат. + предл. пп.

Грамм. обозн.	Кодовые цифры	И н ф о р м а ц и я		
snGD	29	ед.ч. женск.р.	род. + дат.	пп.
snNA	30	" средн.р.	мм. + вин.	пп.
snNAP	31	" "	мм. + вин. + предл.	пп.
plNA	32	множ.ч.	мм. + вин.	пп.
plAG	33	"	вин. + род.	пп.
snGplNA	34	ед.ч. средн.р.	род. + мн.ч.им. + вин.	пп.
smPplNA	35	" мужск.р.	пред. + мн.ч.им. + вин.	пп.
sfGplNA	36	" женск.р.	род. + мн.ч.им. + вин.	пп.
sfGDPplNA	37	" " р.	род. + дат. + предл. + имн. ч. им. + вин.	п.
sAGplN	38	"	вин. + род. + мн.ч.им.	пп.
smGplNA	39	" мужск.р.	род. + мн.ч.им. + вин.	пп.
sfNA	40	" женск.	им. + вин.	пп.
sfGplN	41	" " р.	род. + мн.ч.им.	пп.
snAP	42	" средн.р.	вин. + предл.	пп.
sfPplA	43	" женск.р.	пердл. + мн. п. вин.	пп.

Парадигматические формы глаголов /К I = 28-47/

А/ Формы полных глаголов : К I = 28 - 43, 45 - 47.

При некоторых глаголах в словаре кроме кодового числа К. I. еще имеется т.н. словарное К. II. для обозначения глаголов совершенного и несовершенного вида.

Словарные К. II. для глаголов несовершенного вида = 0.

Словарные К. II. для глаголов совершенного вида = 5.

Кодовые цифры по личным и родовым окончаниям

Грамм. обозн.	Кодовые ч.	Информация
inf.	1	безличная форма
sl	2	ед.ч. 1.л.
s3	3	" 3.л.
pl.1.	4	множ.ч.1.л.
pl3	5	" 3.л.
pm	11	ед.ч. мужск.р. /прошедшее вр./
pf	12	" женск.р.
pn	13	" средн.р.
prl	14	множ.ч.
imp	15	повел.ф. ед.ч. 2.л.

Кодовые цифры по формам глаголов:

1 + сл. К. II.	= 1	безл.форма глагола несов. вида
2 + сл. К. II.	= 2	наст.вр. ед. число 1.л.
3 + сл. К. II.	= 3	" " 3.л.
4 + сл. К. II.	= 4	" мн.ч. 1.л.
5 + сл. К. II.	= 5	" " 3.л.
1 + сл. К. II.	= 6	безличная ф.гл. соверш. вида
2 + сл. К. II.	= 7	будущее простое, ед.ч. 1.л.
3 + сл. К. II.	= 8	" " " 3.л.
4 + сл. К. II.	= 9	простое буд. мн. ч. 1 л.
5 + сл. К. II.	= 10	" " " 3.л.
	11	прош.вр. ед.ч.м.р.
	12	" " ж.р.
	13	" " с.р.
	14	" " множ.ч.
	15	повел.накл. ед.ч. 2.л.
	.	- - - -
	.	- - - -
	56	дееприч. несов.вида
	57	дееприч. сов. вида

б/ Формы вспомогательных глаголов: К.І. = 44

/"есть" входит в состав словаря "полных слов": К.І.44,  
К.ІІ.=3/

Словарный код К.ІІ. у вспомогательных глаголов:

быва	= 0	бывал	= 0
бы	= 0	был	= 0
буд	= 5	стал	= 10
ста	= 15	могл	= 20
стан	= 15 следовал;	приходил	= 30
мог	= 25 представлял;	являл	= 40
мож	= 25 представил;	явил	= 40
приход;	следу	= 35	
предсталя;	явля	= 45	

Кодовые числа парадигматических форм даются как сумма кодовых чисел по окончаниям и соответственных словарных кодов К.ІІ. основ.

1	быть, бывать	неопр. ф.	
2	бываю	наст. вр. ед. ч. 1.л.	
3	бывает	" "	3.л.
4	бываем	"	мн. ч. 1.л.
5	бывают	" "	3.л.
6	- - - -	- - - -	
7	буду	наст. вр. ед. ч. 1.л.	
8	будет	" "	3.л.
9	будем	"	мн. ч. 1.л.
10	будет	" "	3.л.
11	был, бывал	прош. вр. ед. ч. м.р.	
12	-"-а, -"-а	" "	ж.р.
13	-"-о, -"-о	" "	ср.р.
14	-"-и, -"-и	"	множ.ч.
15	- - - -		
16	стать	неопр. ф.	
17	стану	наст. вр. ед.ч. 1.л.	
18	станет	" " "	3.л.



19	станем	наст. множ.ч.	1.л.
20	станут	" "	3.л.
21	стал	прош. вр. ед.ч. м.р.	
22	"-а	" " ж.р.	
23	"-о	" " ср.р.	
24	"-и	" множ.ч.	
25	- - -		
26	мочь	неопр. ф.	
27	могу	нест.вр. ед.ч.	1.л.
28	может	" "	3.л.
29	можем	" множ. ч.	1.л.
30	могут	" "	3.л.
31	мог	прош.вр. ед.ч. мужск.р.	
32	могла	" " женск.р.	
33	"-о	" " средн.р.	
34	"-и	" множеств.ч.	
35	- - -		
36	- - -	- - -	
37	- - -	- - -	
38	приходится, следует	наст.вр.ед.ч.	3.л.
39	" и т.д.	" " мн.ч.	1.л.
40	"	" " "	3.л.
41	приходился, следовал и т.д.	прош.вр. м.р.	
42	" следов	" ж.р.	
43	" "	" с.р.	
44	" "	" множ.ч.	
45	- - -		
46	- - -	- - -	
47	- - -	- - -	
48	представляется является	настоящ. вр. ед.ч.	3.л.
49	- - -	- - -	
50	" "	наст. вр. мн. ч.	3.л.
51	представлялся представился	прош. вр. ед. ч. м.р.	
52	являлся	" " ж.р.	
53	явился и т.д.	" " ср.р.	
54		" множ.ч.	
55	- - -		
56			

Парадигматические формы причастий /К.І. = 48, 53, 54, 55, 56/, прилагательных /К.І. = 49, 50, 51, 52/ относительных местоимений /К.І. = 58/ притяжательных местоимений /К.І.59/, указательных местоимений существительных /К.І. = 60/.

1	мужской род	ед.ч.	И.р.
2	"	"	В.р.
3	"	"	Р.п.
4	"	"	Д.п.
5	"	"	Тв.п.
6	"	"	Пр.п.
7	женский р.	"	И.п.
8	"	"	В.п.
9	"	"	Р.п.
10	"	"	Д.п.
11	"	"	Тв.п.
12	"	"	Пр.п.
13	средний р.	"	И.п.
14	"	"	В.п.
15	"	"	Р.п.
16	"	"	Д.п.
17	"	"	Тв.п.
18	"	"	Пр.п.
19	множеств.ч.		И.п.
20	"		В.п.
21	"		Р.п.
22	"		Д.п.
23	"		Тв.п.
24	"		Пр.п.
25	ед.ч. мужской р.		И + В.
26	" м. + ср.		В + Т.
27	" "		Д.
28	" "		Пр.
29	" ж.р.		Р. + Д. + Тв. + Пр.
30	" ср.р.		И. + В.
31	мн.ч.		В. + Р. + Пр.
32	"		И + В

33	ед.ч.	м + ср.	Тв. + М.ч.Д.
34	сравн.	степень -	простая форма
35	ед.ч.	м.р. И + В. ж.р. Р. + Д. + Тв. + Пр.	
36	"	ж.р. В + Р /её/	
[37	"	" Д + Тв + Пр /ей/]	
38	"	м.р. + ср.р. Тв.	
39	"	ж.р. Р. + Тв.	
45	мн. ч.	В + Р	
46	ед.р.	ж.р. Р. + Д.	

Предикативные формы:

40	ед.ч.	м.р.	
41	"	ж.р.	
42	"	ср.р.	
43	мнот.	ч.	
44	ед.ч.	ср.р. + наречие	

Парадигматические формы личных местоимений /К.І. = 57

1	ед.ч.	1.л.		И.п.
2	"	3.л.	м.р.	И.п.
3	"	"	ж.р.	И.п.
4	"	"	с.р.	И.п.
5	мн.ч.	1.л.		И.п.
6	мн.ч.	3.л.		И.п.
7	ед.ч.	1.л.		В.п.
8	"	3.л.	м + ср.р.	В.п.
9	"	3.л.	ж.р.	В.п.
10	мн.ч.	1.л.		В.п.
11	"	3.л.		В.п.
12	ед.ч.	1.л.		Р.п.
13	"	3.л.	м + ср.р.	Р.п.
14	"	3.л.	ж.р.	Р.п.
15	мн.ч.	1.л.		Р.п.
16	"	3.л.		Р.п.
17	ед.ч.	1.л.		Д.п.
18	"	3.л.	м + ср.р.	Д.п.

19	ед.ч.	3.л.	ж.р.	Д.п.
20	мн.ч.	1.л.		Д.п.
21	"	3.л.		Д.п.
22	ед.ч.	1.л.		Тв.п.
23	"	3.л.	м. + ср.р.	Тв.п.
24	"	3.л.	ж.р.	Тв.п.
25	мн.ч.	1.л.		Тв.п.
26	"	3.л.		Тв.п.
27	ед.ч.	1.л.		Пр.п.
28	"	3.л.	м. + ср.р.	Пр.п.
29	"	3.л.	ж.р.	Пр.п.
30	мн.ч.	1.л.	ж.р.	Пр.п.
31	"	3.л.		Пр.п.
32	ед.ч.	3.л.	м. + ср.р.	В + Р.пп.
33	мн.ч.	1.л.		В + Р.пп.
34	ед.ч.	3.л.	м. + ср.р.	В + Р.пп. + прит.местоим.
35	"	3.л.	ж.р.	В + Р.пп. + прит.местоим.
36	"	3.л.	ж.р.	В + Р.пп.
37	мн.ч.	3.л.		В + В.пп. + прит.ми.
38	"	3.л.		В.Р. + Пр.п.
39	ед.ч.	3.л.	м. + ср.р.	Тв.п./с предло- гом, мн.ч. 3.л. Д.п.
40	"	3.л.	м. + ср.р.	Тв.п./без предлога/ мн.ч. 3.л. Д.п.
41	"	3.л.	ж.р.	Тв + Д.пп.
42	"	3.л.	ж.р.	Тв + Д + Пр.пп.
				"
				"
				"
				"
				"
				"
				"
				"
				"

Парадигматические формы неопределенных и количественных числительных, К.І.=61.

- 1 - 33 совпадает с кодовыми числами для определений
- |    |             |             |        |
|----|-------------|-------------|--------|
| 34 | ед.ч.       | м. + ср.р.  | И.В.п. |
| 35 | И + В.      | пп.         |        |
| 36 | В + Р + Пр. | пп.         |        |
| 37 | Д.          | п.          |        |
| 38 | Тв.         | п.          |        |
| 39 | м.р.        | И.п.        |        |
| 40 | ср.р.       | И.п.        |        |
| 41 | м.р.        | В.п.        |        |
| 42 | ср.р.       | В.п.        |        |
| 43 | ж.р.        | И + В       | пп.    |
| 44 | "           | И.п.        |        |
| 45 | "           | В.п.        |        |
| 46 | Р.          | п.          |        |
| 47 | Пр.         | п.          |        |
| 48 | Р + Д 2 Тв. | + Пр.       | пп.    |
| 49 | И.          | п.          |        |
| 50 | В.          | п.          |        |
| 51 | В + Р.      | пп.         |        |
| 52 | ед.ч.       | Тв. + мн.ч. | Д.п.   |

"Парадигматические" формы модальных слов к сказуемому

К.І. = 62

В словаре т.н. "полных слов": можно, надо, нужно,  
необходимо К.І. = 62;  
К.ІІ. = I.

/слова: "нужн" и "необходим" имеются и в словаре основ как прилагательные/

В словаре основ: должен, должн, словарный К.ІІ. = 0  
 способен, способн " = 4  
 годен, годн " = 8

Кодовые числа окончаний: мужск. р. = 2  
 ж.р. = 3  
 с.р. = 4  
 м.н.ч. = 5

Кодовые числа парадигматических форм получаются как сумма кодовых чисел окончаний и словарного кода К.ІІ.

2	}	формы от "должен"
3		
4		
5	}	формы от "способен"
6		
7		
8	}	формы от "годен"
9		
10		
11	}	формы от "нет"
12		
13		
14	"нет"	
15	"бы"	
16	"чтобы"	

"Парадигматические" формы для слов смешанного класса

/К.І. = 63/:

1	предлог с управлением	Р. п.-а
2	" "	Д. п-а
3	" "	В. п-а
4	" "	Тв.п-а
5	" "	Пр.р-а
6	" "	В и Тв. пп.

7	предлог с управлением	В и Пр. пп.
8	" "	Д.и Пр. пп.
9	" "	Р. и Тв. пп.
10	наречия /не на-о и -ски/	
11	"не"	
12	точка	
13	запятая	
14	точка с запятой	
15	"и", "или"	
16	"а", "но"	
17	союзы /которые ни в русском, ни в венгерском языках не дают особой конструкции/	
18	_____	
19	_____	
20	возвратное местоимение	В.п.
21	" "	Р.п.
22	" "	Д.п.
23	" "	Тв.п.
24	" "	Пр.п.
25	" "	В + Р. пп.
26	" "	Д + Пр.пп.
27	- - - -	
28	- - - -	
29	- - - -	
30	начало предложения	
31	междометие	
32	выражение	
33	формула	
34	сокращение	
35	гомонимная форма	

Информации о слове как о самостоятельной единице, разделенной на основу и окончание, получается сопоставлением информации его частей. Машинное решение этого вопроса особенной трудности не представляет. Операции могут быть выражены следующими формулами:

1/ Для слов на - а:

$$\begin{aligned} & [ /S-1 \vee S-13 \vee S-14 / + -a ] \vee [ /S-2 \vee S-15 / + -a ] \vee [ /S-5 \vee \\ & \vee S-6 / + a ] \vee [ S-9 + -a ] \vee [ S-11 + -a ] \vee [ S-12 + -a ] \vee \\ & \vee [ /V-1---V18 \vee V-19---V-21 / + -a ] \vee [ Vau + -a ] \vee [ P-5 + \\ & + -a ] \vee [ /A-1 \vee A-2 / + -a ] \vee [ P-3 + -a ] \vee [ Pn-1 + -a ] \vee \\ & \vee [ Pn-4 + -a ] = \end{aligned}$$

$$\begin{aligned} & = S-msG \vee S-smAG \vee S-fsN \vee S-snGplNA \vee S-smGplNA \vee \\ & \vee S-nsG \vee V-pf. \vee Vau-pf. \vee P-5sf. \vee A-1,2sf. \vee P-3sf \vee \\ & \vee Pn-1s3fN \vee Pn-4 sf. N \end{aligned}$$

2/ Для слов на -ов, -ев:

$$\begin{aligned} & [ /S-1 \vee S-3 \vee S-14 / + /-ов \vee -ев / ] \vee [ /S-2 \vee S-15 / + /-ов \vee \\ & \vee -ев / ] = S-plG \vee S-plAG \end{aligned}$$

3/ Для слов на -ей:

$$\begin{aligned} & [ /S-1 \vee S-7 \vee S-9 \vee S-14 \vee S-16 / + -ей ] \vee [ /S-2 \vee S-15 / + -ей ] \vee \\ & \vee [ /S-5 \vee S-6 \vee S-8 / + -ей ] \vee [ /A-2 \vee A-3 / + -ей ] \vee \\ & [ /P-1 \vee P-2 / + -ей ] \vee [ Pn-3 + -ей ] = \end{aligned}$$

$$\begin{aligned} & = S-pl G \vee S-pl AG \vee S-fsI \vee A-2,3-s fGDIPr \vee P-1,2-sfGDIPr \vee \\ & \vee Pn-3-sfGDIPr \end{aligned}$$



3. Определение точных парадигматических форм -  
- согласованные номинальные конструкции

После сопоставления основ и окончаний для определения парадигматических форм еще остается довольно много грамматических многозначных форм существительных. Для выявления действительной однозначной парадигматической формы прибегаем к сопоставлению информации грамматически согласованных форм имен с прилагательными и предлогами. Известно, что это представляет иногда трудную задачу даже для человека, потому что слова, сочтенные согласованием, могут расположиться в разнообразных, иногда довольно сложных структурах. Приступить к анализу согласованных членов какой-то номинальной конструкции без предыдущего выяснения структуры данной конструкции, было бы очень отважным делом. Поэтому вместе с предварительным анализом слов номинальной конструкции мы стремимся сразу определять и структуру, в которую входят слова, и только после этого производится выяснение парадигматической формы.

Для выражения самых распространенных номинальных структур можно применять следующую формулу:

$$/Pr \ v \wedge / + /Av \wedge / + /Pnv \wedge / + /C \ v \wedge / + /P \ v \wedge / + Nu \ v \wedge / + S \quad (1)$$

Новые обозначения:  $v$  = или

$\wedge$  = пустой класс

Эта формула содержит очень многообразный распорядок; члены могут повторяться и располагаться в любом возможном порядке. Так в формуле (1) содержатся напр.:

A + A + S

Pr + A + S

Pr + A + C + A + S

Nu + A + S и т.д.

Наречий в структуре нет. Они с точки зрения структуры никакой роли не играют и поэтому не входят в структуру номинальных конструкций.

Для обнаружения согласованных номинальных конструкций мы имели в виду и более сложные структуры. Структура, распространенная причастным оборотом, имеет вид:

$$\begin{aligned}
 & /Pr_{O} \vee \wedge / + A'_{O} \vee \wedge / + PA_{O} \vee \wedge / + S_{O} \vee \wedge / + Pr_{PA_{O}} \vee Pr_{S} \vee \wedge / + \\
 & + A'_{PA_{O}} \vee A'_{S} \vee \wedge / + S_{PA_{O}} \vee S_{S} \vee \wedge / + Pr_{S/S} \vee \wedge / + S_{S/S} \vee \wedge / + \\
 & + A'_{O/PA_{O}} \vee / + S_{O/PA_{O}} \vee \wedge \quad (2)
 \end{aligned}$$

- где:
- $Pr_{O}$  - предлог, открывающий номинальную конструкцию;
  - $A'$  -  $(A \vee \wedge) + (Pn \vee \wedge) + (Nu \vee \wedge) + (C \vee \wedge) + (P \vee \wedge)$  с индексами или без индексов;
  - $A'_{O}$  -  $A'$  и входит в одну конструкцию с  $S_{O}$ ;
  - $PA_{O}$  - причастие /или прилагательное/ с распространением;
  - $Pr_{PA_{O}}$  - предлог с управляющим словом  $PA_{O}$ ;
  - $Pr_{S}$  - предлог с управляющим словом  $S$  /перед него/;
  - $A'_{PA_{O}}$  -  $A'$  с управляющим словом  $PA_{O}$ ;
  - $A'_{S}$  -  $A'$  с управляющим словом  $S$  /перед него/;
  - $S_{PA_{O}}$  - существ. с управляющим словом  $PA_{O}$ ;
  - $S_{S}$  - существ. с управляющим словом  $S$  /перед него/;
  - $Pr_{S/S}$  - предлог с управляющим словом  $S_{PA_{O}}$ ;
  - $A'_{S/S}$  -  $A'$  управляемое  $S_{PA_{O}}$ ;

- $S_{s/s/}$  - существ., управляемое  $S_{PA_0}$ ;
- $A'_0/PA_0/$  -  $A'$ , согласуется вместе с  $PA_0$  с существительным  $S_0/PA_0/$ ;
- $S_0/PA_0/$  -  $S$ , образующее вместе с  $PA_0$  "рамку".

Такая структура разбивается на следующие главные части:

- а/ номинальную группу, факультативно содержащую причастие или прилагательное с распространением, т.е.:

$$/Pr_0 \vee \wedge + /A'_0 \vee \wedge + /PA_0 \vee \wedge + S_0 ;$$

- б/ номинальную группу, образующую распространение причастия  $PA_0$ ;

$$/Pr_{PA_0} \vee \wedge + /A'_{PA_0} \vee \wedge + S_{PA_0} ;$$

- в/ номинальную группу, дающую т.н. несогласованное опеределение к существительному  $S_{PA_0}$ ;

$$/Pr_{s/s/} \vee \wedge + /A'_{s/s/} \vee \wedge + S_{s/s/}$$

Если конструкция а, действительно содержит член  $PA_0$ , то существительное  $S_0$  /может быть и  $A'_0/$  образуют "рамку", внутри которой поместятся конструкции б, и в, /существительное в этом случае обозначается  $S_0/PA_0/$  /. Такая расстановка членов не вводит никакого изменения ни в число групп, ни в их характер.

Число групп не изменяется и тогда, если из групп б, или в, имеется больше, чем одна. В этом случае новая группа вступает на место законченной и тождественной по характеру предыдущей.

Для анализа сложных номинальных конструкций нам придется принимать еще одну структуру, характерная черта которой придается также причастным оборотом. Она отличается от предыдущей тем, что существительное, управляющее причастием, стоит перед последним и отделяется от него запятой. Схема этой структуры:

$$S_0 + C + PA_0 + S_{PA_0} + C$$

здесь  $PA_0$  -  $PA$  с управляющим  $S_0$ , и стоящим после него.

Более полная форма этой структуры:

$$\begin{aligned} & /Pr_0 \vee \wedge / + /A'_0 \vee \wedge / + S_0 + /Pr_S \vee \wedge / + /A'_S \vee \wedge / + S_S \vee \wedge / + C + \\ & + PA_0 + /Pr_{PA_0} \vee \wedge / + /A'_{PA_0} \vee \wedge / + S_{PA_0} + /Pr_S \vee \wedge / + /A'_S \vee \wedge / + \\ & + /S_S \vee \wedge / + C \end{aligned} \quad (3)$$

Изображаем только самые важные члены:

$$S_0 + /S_S/ + C + PA_0 + S_{PA_0} + /S_S/ + C .$$

Расчленение номинальных структур и обособление главных групп интересно для нас не только с точки зрения теоретического изучения фразы, но и для определения и выбора тех членов, которые необходимо нужны для дальнейшего анализа всей структуры. Анализ всех номинальных конструкций производится на основе такой структурной схемы с незаполненными местами для отдельных членов.

Соединяя структуры /2/ и /3/, мы получаем:

$$\begin{aligned} & /Pr_0 \vee \wedge / + /A'_0 \vee \wedge / + S_0 \vee \wedge / + /Pr_S \vee \wedge / + /A'_S \vee \wedge / + S_S \vee \wedge / + C \vee \wedge / + \\ & + /Pr_0 \vee \wedge / + /A'_0 \vee \wedge / + PA_0 \vee \wedge / + S_0 \vee \wedge / + /Pr_{PA_0} \vee \wedge / + Pr_{PA_0} \vee \wedge / + \\ & \vee Pr_S \vee \wedge / + /A'_{PA_0} \vee \wedge / + A'_{PA_0} \vee \wedge / + /A'_S \vee \wedge / + S_{PA_0} \vee \wedge / + S_{PA_0} \vee \wedge / + S_S \vee \wedge / + \\ & + /Pr_{S_{PA_0}} \vee \wedge / + Pr_{S_{PA_0}} / \vee \wedge / + /A'_{S_{PA_0}} \vee \wedge / + A'_{S_{PA_0}} / \vee \wedge / + /S_{S_{PA_0}} \vee \wedge / + S_{S_{PA_0}} / \vee \wedge / + \\ & \vee \wedge / + /A'_0 / PA_0 / \vee \wedge / + S_0 / PA_0 / \vee \wedge / + C \vee \wedge / . \end{aligned}$$

Принимая, что  $A'' = /Pr \vee \wedge + A' /$ , и оставляя только те члены, которые нужны для машинного анализа, получаем:

$$\begin{aligned}
& /S_0 \vee \wedge / + /S_S \vee \wedge / + /C \vee \wedge / + /PA_0 \vee \wedge / + /A''_0 \vee A''_{PA_0} \vee \wedge / + /PA_0 \vee \wedge / + \\
& + /S_0 \vee S_{PA_0} \vee /A''_{PA_0} \vee A''_S \vee \wedge / + /S_{PA_0} \vee S_S \vee \wedge / + /A''_S / \varepsilon / \vee \wedge / + \\
& + /S_S / \varepsilon / \vee \wedge / + /A''_0 / PA_0 / \vee \wedge / + /S_0 / PA_0 / \vee \wedge /
\end{aligned}$$

Поставим теперь члены  $S_0/PA_0/$  и  $A''_0/PA_0/$ , дающие рамочную структуру, к члену  $PA_0$ , получаем окончательную формулу для анализа русских номинальных конструкций:

$$\begin{aligned}
& /S_0 \vee \wedge / + /S_S \vee \wedge / + /C \vee \wedge / + /PA_0 \vee \wedge / + + \\
& + /A''_0 \vee A''_{PA_0} \vee A''_{PA_0} \vee \wedge / + /PA_0 \vee \wedge / + /S_0 \vee S_{PA_0} \vee S_{PA_0} \vee \wedge / + + \\
& + /A''_{PA_0} \vee A''_S \vee \wedge / + /S_{PA_0} \vee S_S \vee \wedge / + \\
& + /A''_S / \varepsilon / \vee \wedge / + /S_S / \varepsilon / \vee \wedge /
\end{aligned} \tag{4}$$

#### 4. Редукция грамматической многозначности членов номинальной конструкции до однозначности

Редукция грамматической многозначности членов номинальной конструкции до однозначности производится в МП р - в параллельно со структурным анализом упомянутых номинальных групп. Грамматически многозначные парадигматические формы имен дают достаточное число информации для того, чтобы правильно поместить их в структурной схеме номинальных групп. Расчленение структуры /4/ дает возможность присоединять каждому существительному "свои" определения, а сопоставление сочтенных многозначных форм допускает действительное существование только некоторых грамматических значений. Это и означает редукцию многозначности.

Несколько формул для редукции грамматической многозначности.

1/ Для существительных S msNA

$$\left\{ \left[ \begin{array}{l} /PrAvPrA| \ v \ PrAP \ v \wedge / + /Pn \ ms \ NA \ v \wedge / + /Nu \ ms \ NA \ v \wedge / + \\ + /A \ msNA \ v \wedge / \end{array} \right] + S \ msNA \right\} =$$

$$= \left\{ \left[ \begin{array}{l} PrA + /Pn \ msA \ v \wedge / + /Nu \ ms \ A \ v \wedge / + /A \ msA \ v \\ v \wedge / \end{array} \right] + S \ msA \right\} \ v \ \left\{ \left[ \begin{array}{l} /Pn \ msNA \ v \wedge / + /Nu \ msNA \ v \wedge / + \\ + /AmsNA \ v \wedge / \end{array} \right] + SmsNA \right\}$$

2/ Для существительных s m s DP

$$\left\{ \left[ \begin{array}{l} /Pr \ D \ v \ Pr \ DP \ v \ PrP \ v \ PrAP \ v \wedge / + /Pn \ smnD \ v \ Pn \ smnP \ v \\ v \wedge / + /NusmnD \ v \ NusmnP \ v \wedge / + /AsmnD \ v \ AsmnP \ v \wedge / \end{array} \right] + \right.$$

$$\left. + SmsDP \right\} =$$

$$= \left\{ \left[ \begin{array}{l} /PrD \ v \wedge / + /PnsmD \ v \wedge / + /NusmDv \wedge / + /AsmDv \wedge / \end{array} \right]^x + SmsD \right\} \ v$$

внутри скобки [ ]<sup>x</sup> должен быть член с действительным значением

внутри скобки [ ]<sup>x</sup> должен быть член с действительным значением

$$\left\{ \left[ \begin{array}{l} /PrP + /Pn \ smP \ v \wedge / + /NusmP \ v \wedge / + /AsmP \ v \wedge / \end{array} \right] + SmsP \right\} \ v$$

$$\left\{ \left[ \begin{array}{l} /Pr \ DP / + SsmDP \end{array} \right] \right\}$$

3/ Для существительного SsmAG:

$$\left\{ \left[ \begin{array}{l} /PrA \ v \ PrA \ v \ PrAP \ v \ PrG \ v \ PrG \ v \wedge / + /Pn \ smnAG \ v \ Pn \ plAG \ v \\ v \ Pn \ pl \ NA \ v \wedge / + /Nu \ smnAG \ v \ Nu \ smnNA \ v \ NuNA \ v \wedge / + \\ + /AsmnAG \ v \ A \ pl \ AG \ v \wedge / \end{array} \right] + SsmAG \right\} =$$

$$\begin{aligned}
&= \left\{ \left[ /PrA + /Pn\ smA\ v\ \wedge/+ /NusmA\ v\ \wedge/+ /AsmA\ v\ \wedge/ \right] + \right. \\
&+ SsmA \left. \right\} v \\
&v \left\{ \left[ /Pn\ smnAG\ v\ \wedge/+ /Nu\ smnAG\ v\ \wedge/+ /AsmnAG\ v\ \wedge/ \right] + \right. \\
&v SsmAG \left. \right\} v \\
&v \left\{ \left[ /Pr\ G + /PnsmG\ v\ \wedge/+ /Nu\ smG\ v\ \wedge/+ /A\ s\ m\ G\ v\ \wedge/ \right] + \right. \\
&+ SsmG \left. \right\} v \\
&v \left\{ \left[ /Pn\ pl\ G\ v\ Ph\ pl\ N\ v\ \wedge/+ Nu\ m\ N + /A\ pl\ G\ v\ \wedge/ \right] + \right. \\
&+ SsmG \left. \right\}
\end{aligned}$$

/G' обозначает родительный надеж в случае числительного в именительном надеже/

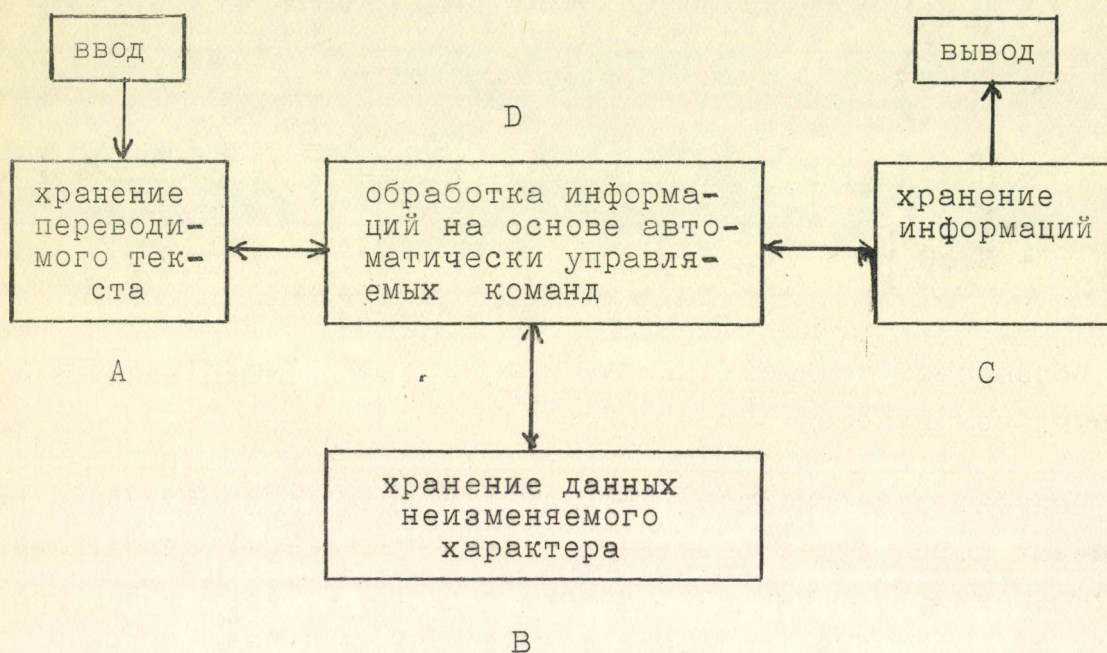
Подобно этим имеются формулы для всех грамматически многозначных форм существительных, прилагательных, числительных и местоимений, которые расположены в форме таблицы<sup>x</sup>. Несмотря на то, что таблицы содержат данные формул в виде кодовых чисел, все-таки дадут они наглядную картину о возможностях редукции многозначных парадигматических форм.

Приведенные формулы очень похожи на конъюнкционные и дизъюнкционные формулы математической логики, которые могут быть перестроены по правилу дистрибуции. То же самое делается в наших формулах, но не по формальным правилам логики, а учитывая грамматические возможности русской номинальной конструкции [8].

<sup>x</sup> Большинство этих таблиц было составлено Г.Феници и главным образом, Г.Шипеци, который разработал и программу для использования таблиц при решении многозначности.

### III. ПРОГРАММА МП Р - В

#### 1. Схематический план машинных операций



В части "С" машины должно быть обеспечено место для следующего:

- 1/ Хранение информации о всех словах одного предложения;
- 2/ Хранение "рабочих данных" о составляющих номинальных структур;
- 3/ Хранение данных о релевантных словах, опережающих "рабочее слово".



1/ Информационные данные о словах предложения содержат:

- INF - 1 = адрес русского слова,
- INF - 2 = первый код /К.I./ русского слова,
- INF - 3 = второй код /К.II./ русского слова,
- INF - 4 = третий код /К.III./ русского слова,
- INF - 5 = адрес венгерского эквивалента.

2/ Для хранения составляющих номинальных структур требуются:

1/ рабочие данные	o A"	} /первый член из составляющих A"/
2/ " "	Pr	
3/ " "	A	
4/ " "	Pn	
5/ " "	Nu	
6/ " "	A' o(PA <sub>o</sub> )	
7/ " "	A"-U	} индекс при каждом члене группы: = o
8/ " "	PA <sub>o</sub>	
9/ рабочие данные	o A"	
10/ " "	Pr	
11/ " "	A	
12/ " "	Pn	
13/ " "	Nu	
14/ " "	A"-U	
15/ " "	S	
16/ рабочие данные	o A"	} индекс при каждом члене группы: = s(s)
17/ " "	Pr	
18/ " "	A	
19/ " "	Pn	
20/ " "	Nu	
21/ " "	A"-U	
22/ " "	S	

В "рабочие данные" входят:

- 1/ или: а = адрес информации о слове,
- 2/ или: б = первый код /К.І./ слова,
- 3/ или: с = второй код /К.ІІ./ слова,
- 4/ или: д = третий код /К.ІІІ./ слова.

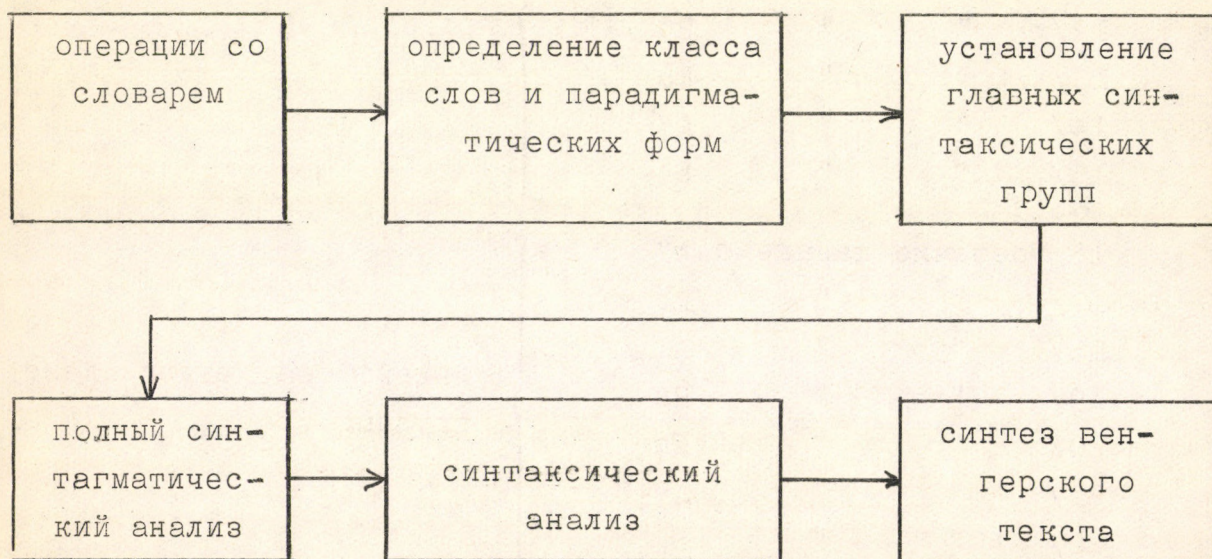
3/ Релевантные слова хранятся в ячейках Q, X, Y, Z, причем их расположение относительно к рабочему слову:

Z + Y + X + Q + "рабочее слово"

Релевантные слова выбираются программой, и о них фиксируются рабочие данные.

## 2. Блок-схема перевода

Самые важные группы операций МП р - в могут быть представлены в следующей схеме:



Такое распределение операций МП в главных чертах совпадает со схемой операций, применяемых в других группах машинного перевода. Особой чертой является выполнение предвари-

тельного синтаксического анализа до проведения синтагматического. Нам это кажется целесообразным, так как обнаружение словосочетаний в предложении таким образом производится легче. Предварительный синтаксический анализ выполняется на основе анализа интерпункционных знаков и отрезков предложения. При этом используются слова, служащие предикатом в предложении и снабженные для облегчения анализа особыми кодовыми числами. /Эта часть алгоритма МП р - в еще не разработана полностью./

### 3. Операции со словарем

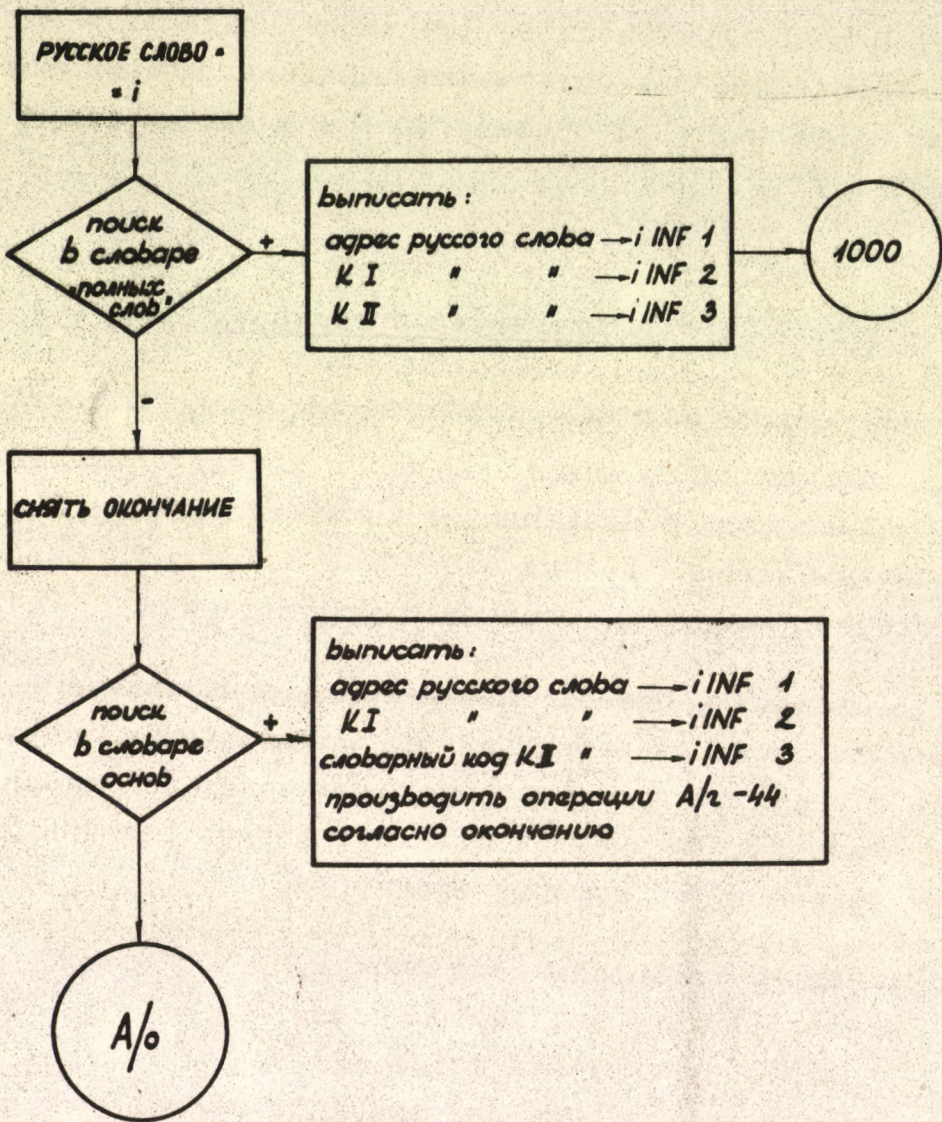
При разработке операций со словарем мы ограничивались пока поиском слов. Для того, чтобы избежать удаления последних букв основы, совпадающих с каким-то окончанием, временно применяем словарь "полных слов", по составу которого проверяются все слова, имеющие соответствующую длину.

В словаре у основы помещены кодовые числа: К.І. и /для некоторых слов/ словарный К.ІІ. По окончании операций со словарем и редуций грамматического значения до однозначности переводимое слово имеется только в форме различных информационных чисел, в ряд которых входят:

- 1/ адрес слова языка-источника,
- 2/ морфологический класс слова,
- 3/ парадигматическая форма слова,
- 4/ синтагматические информации о слове,
- 5/ адрес венгерского эквивалента.

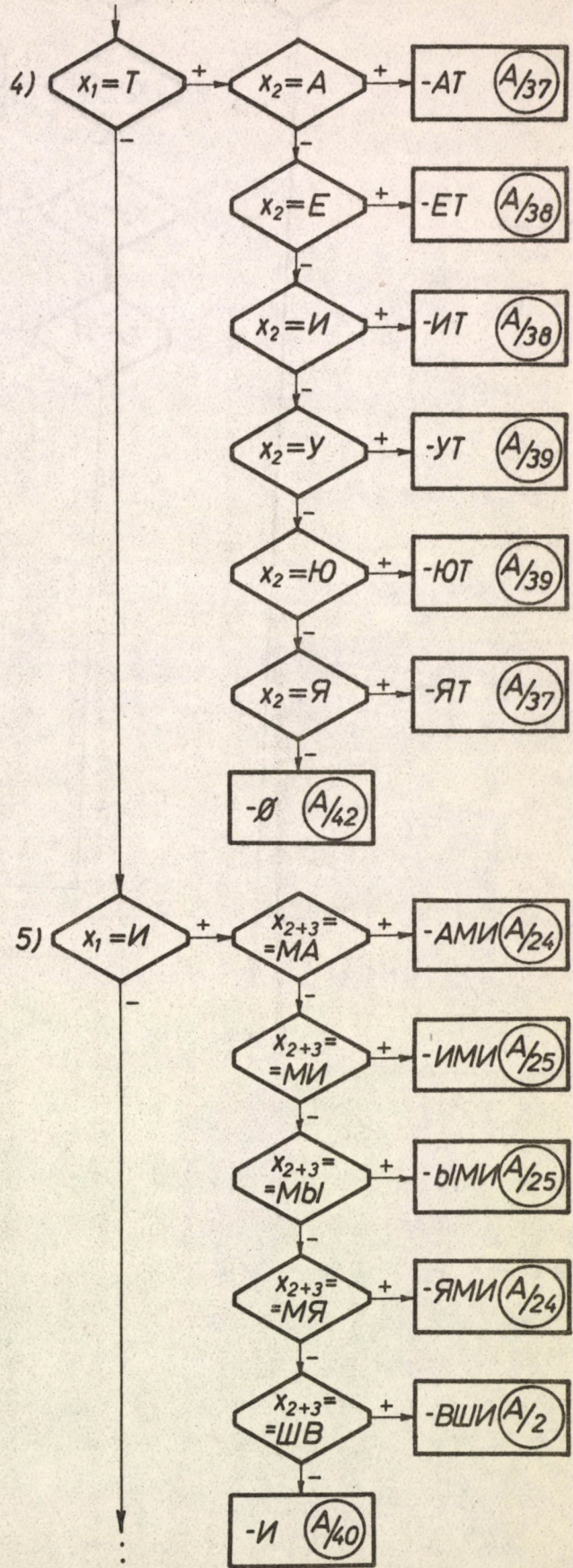
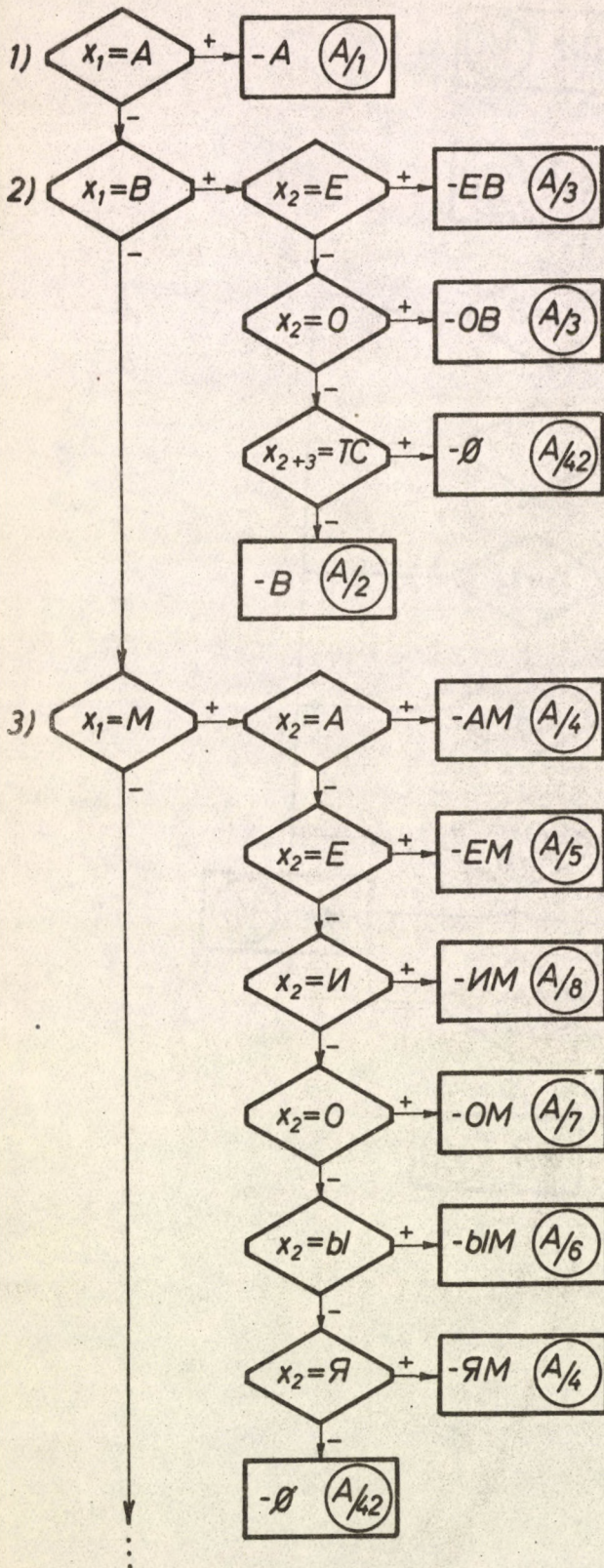
К отделению окончаний и сопоставлению информации от окончания и основы присоединяется несколько операций, которые служат приготовлением к дальнейшей обработке слова. Приготовление состоит в том, что на основе уже полученных информации изменяются параметры некоторых операторов. /2/І, 2/ІІ, 2/ІІІ служат для согласования членов номинальной группы, с помощью В/ІV. найдутся причастные обороты, стоящие за управляющим словом./ Число после "U" указывает на номер следующей операции.

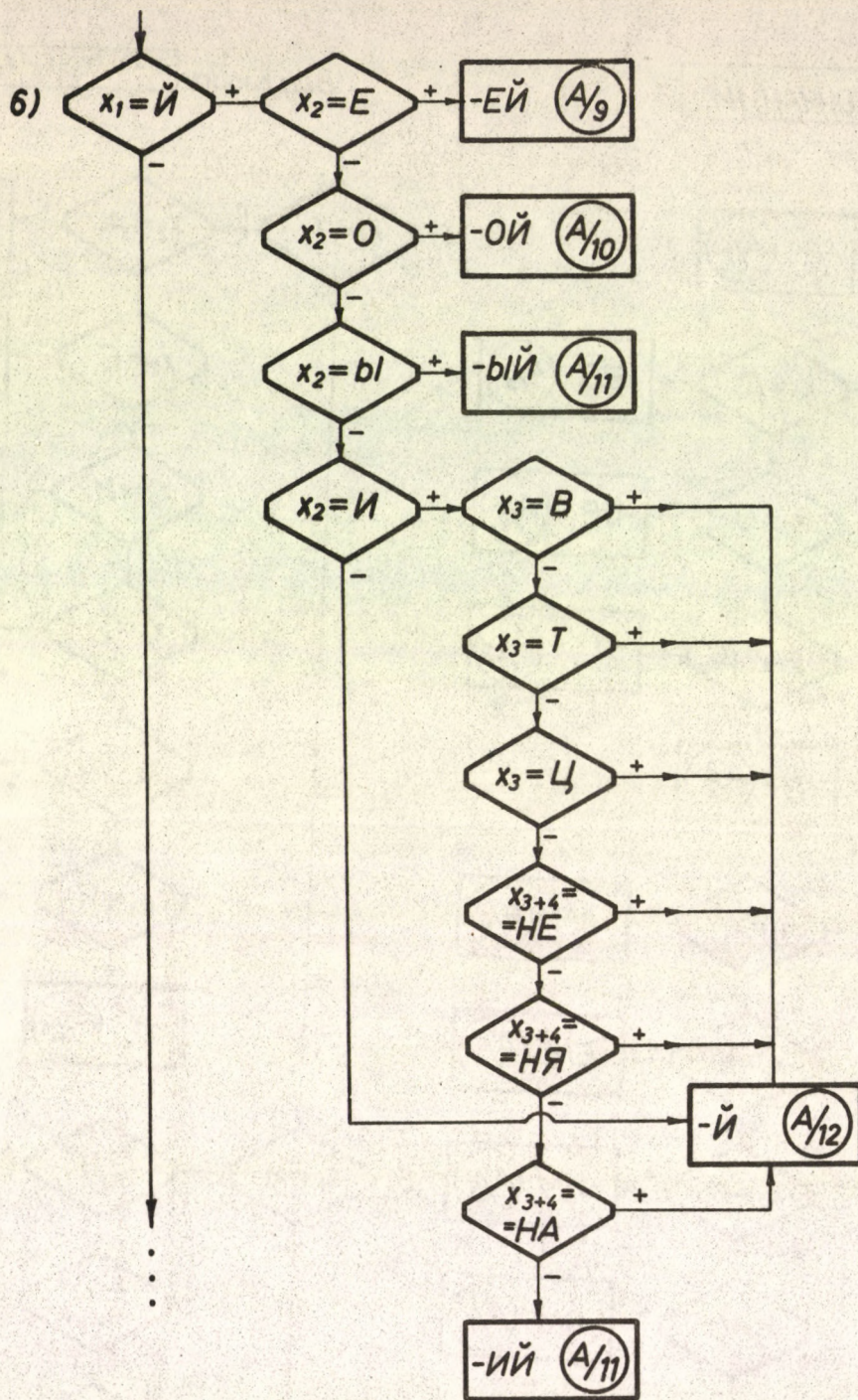
**НАЧАЛО ПРОГРАММЫ**

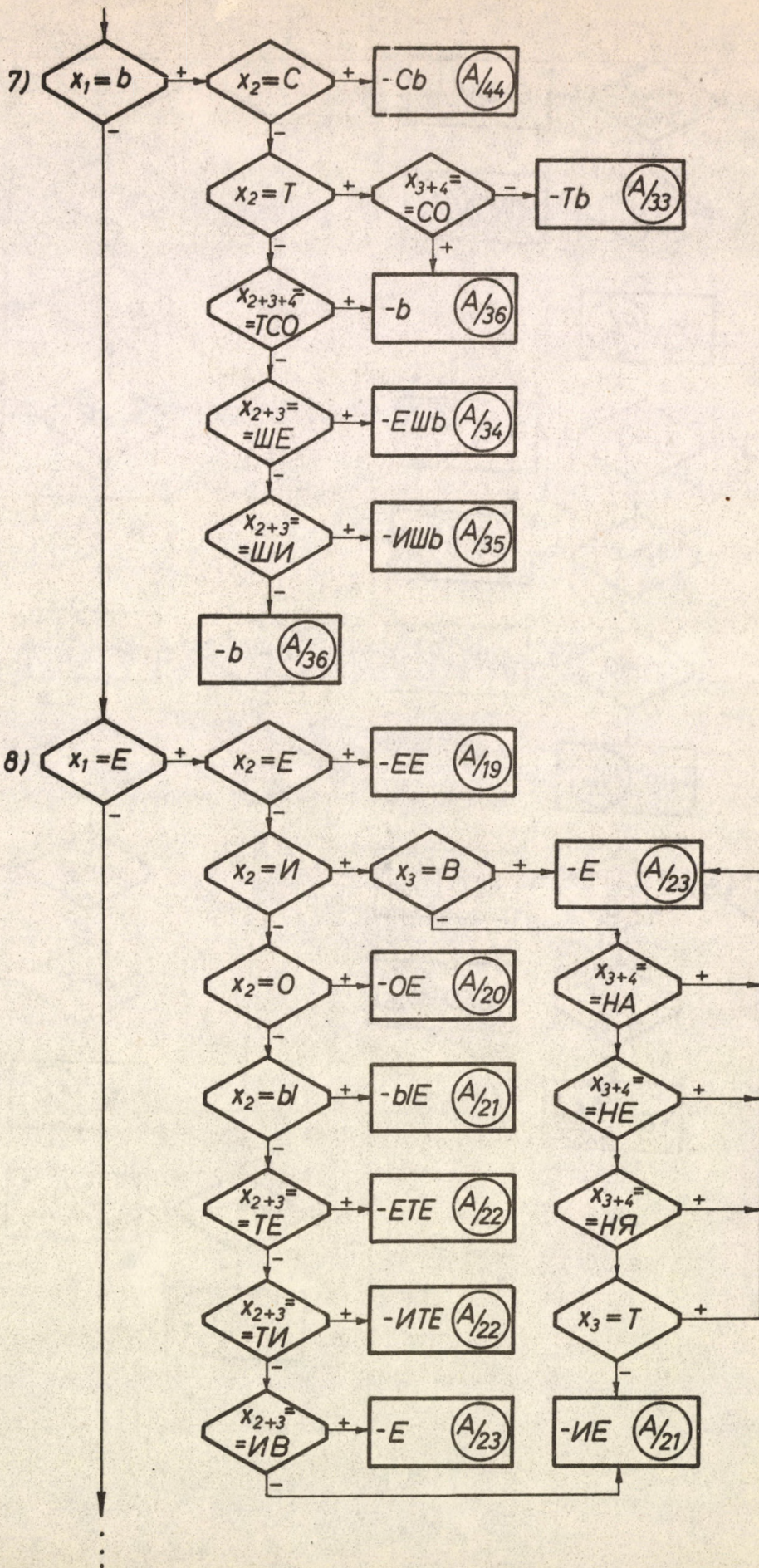


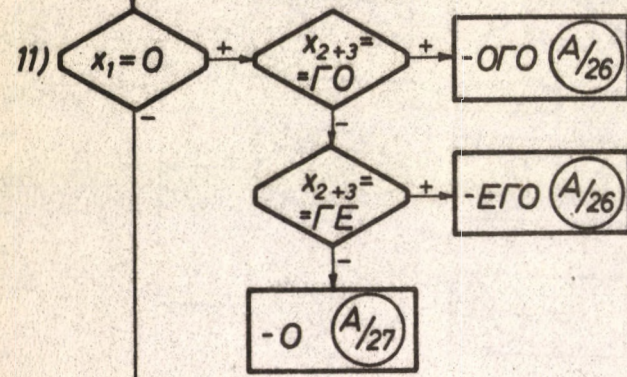
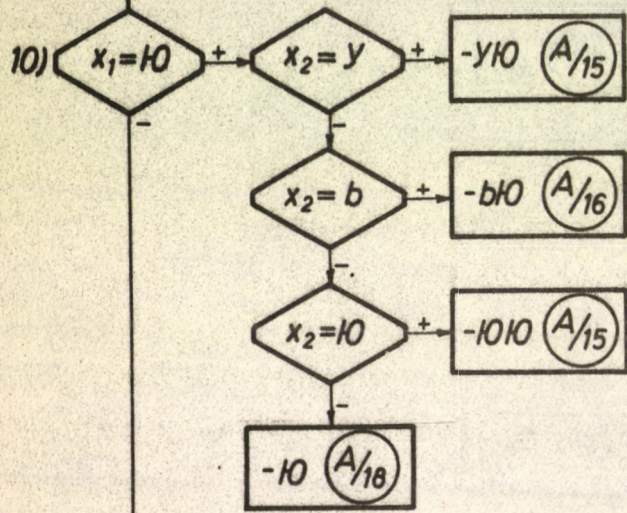
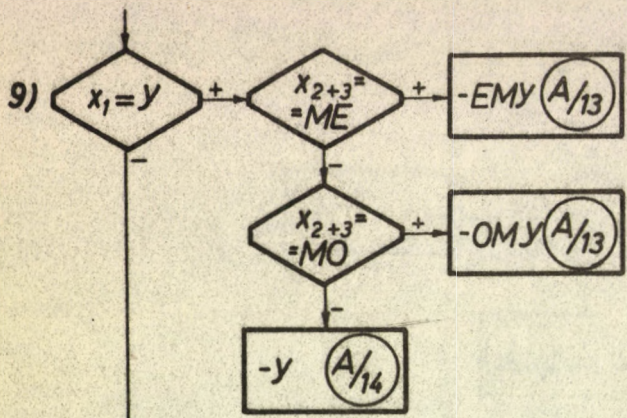
# УДАЛЕНИЕ ОКОНЧАНИЙ

Вид слова:  $\cdot \cdot x_6 x_5 x_4 x_3 x_2 x_1$

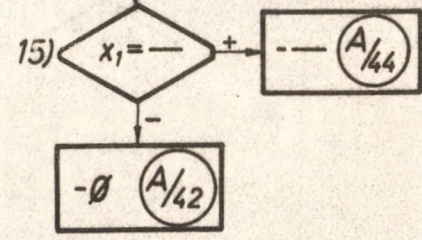
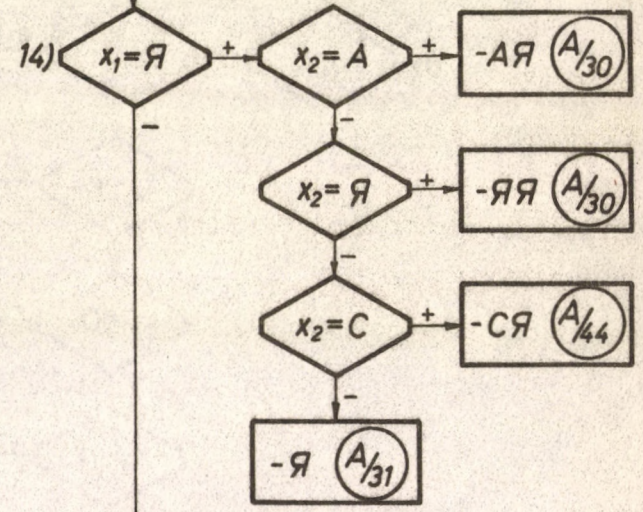
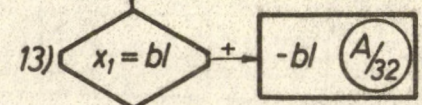
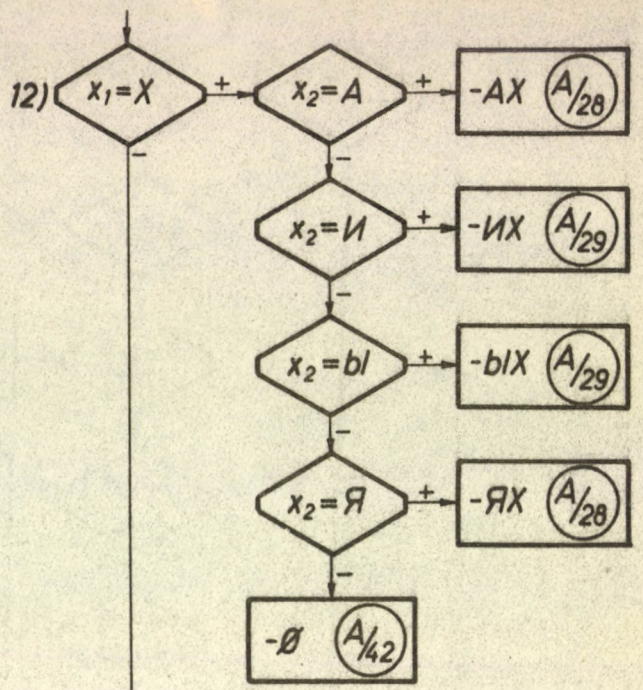








⋮





iINF 2	1,13,14	2,15	5,6	9	11	12	28-43 45-47	44	48	49-50	55	57	60
→ iINF 3	3	27	7	34	39	15	12	12+ak	41	41	41	3	7
→ 2/I.	B 3/I	B.27/I	-	B 34/I	B 39/I	B 15/I	-	-	-	-	-	-	-
→ 2/II.	B 3/II	B 27/II	-	B 34/II	B 39/II	B 15/II	-	-	-	-	-	-	-
→ 2/III.	B 3/III	B 27/III	B 7/III	B 34/III	B 9/III	B 15/III	-	-	-	-	-	-	C 7/III
→ B/IV.	-	-	-	-	-	-	-	-	-	-	-	-	B 7/IV
U:	230	230	230	230	230	230	240	240	240	1005	210	230	190

A/2 - ВШИ

i INF 2	28-47
→ i INF 3	57
→ 2/I	-
→ 2/II	-
→ 2/III	-
→ B/IV	-
U:	240

A/3 - -OB, -EB

i INF 2	1,3,14	2,15
<del>i INF 3</del>	21	33
<del>2/I</del>	B 21/I	B 33/I
<del>2/II</del>	B 21/II	B 33/II
<del>2/III</del>	B 21/III	B 33/III
→ B/IV	-	-
U:	230	230

A/4 - -AM, -AM

i INF 2	1 - 7
<del>i INF 3</del>	22
<del>2/I</del>	B 22/I
<del>2/II</del>	B 22/II
<del>2/III</del>	B 22/III
<del>B/IV</del>	-
U:	230

A/5 - EM

i INF 2	1,2,3	14 - 17	9,10	28-47	51	53, 54	59	60
→ iINF 3	5	5	17	4+sk	28	28	28	33
→ 2/I	B 5/I	B 5/I	B 17/I	-	C 28/I	C 28/I	C 28/I	C 33/II
→ 2/II	B 5/II	B 5/II	B 17/II	-	C 28/II	C 28/II	C 28/II	C 33/II
→ 2/III	B 5/III	B 5/III	B 17/III	-	-	-	-	C 33/III
→ B/IV	-	-	-	-	B 28/IV	B 28/IV	B 28/IV	B 33/IV
U:	230	230	230	240	190	210	190	190

A/6 - 4M

A/7 - OM

i INF 2	49-50	55-56	58
→ i INF 3	33	33	33
→ 2/I	C33/I	C33/I	C33/I
→ 2/II	C33/II	C33/II	C33/II
→ 2/III	C33/III	C33/III	C33/III
→ B/IV	B33/IV	B33/IV	B33/IV
U:	1005	210	190

i INF 2	1,2,11,13,14,15	9,12	<del>49</del> -50	55-56	58	60	61
→ i INF 2	5	17	28	28	28	28	28
→ 2/I	B 5/I	B17/I	C28/I	C28/I	C28/I	C28/I	C28/I
→ 2/II	B 5/II	B17/II	C28/II	C28/II	C28/II	C28/II	C28/II
→ 2/III	B 5/III	B17/III	-	-	-	-	-
→ B/IV	-	-	-	B28/IV	B28/IV	B28/IV	B28/IV
U:	230	230	1005	210	190	190	190

i INF 2	28 - 47	49 - 51	53 - 54	59	60	61
→ i INF 3	4 + sk	33	33	33	33	33
→ 2/I	-	C 33/I	C 33/I	C 33/I	C 33/I	C 33/I
→ 2/II	-	C 33/II	C 33/II	C 33/II	C 33/II	C 33/II
→ 2/III	-	C 33/III	C 33/III	C 33/III	C 33/III	C 33/III
→ B/IV	-	B 33/IV	B 33/IV	B 33/IV	B 33/IV	B 33/IV
U:	240	190	210	190	190	190

A/9 - ЕИ

i INF 2	1,7,9,14,16	2,15	5,6,8	50,51	53,54	59
→ i INF 3	21	33	11	29	29	29
→ 2/I	B 21/I	B 33/I	B 11/I	C 29/I	C 29/I	C 29/I
→ 2/II	B 21/II	B 33/II	B 11/II	C 29/II	C 29/II	C 29/II
→ 2/III	B 21/III	B 33/III	B 11/III	C 29/III	C 29/III	C 29/III
→ B/IV	-	-	-	B 29/IV	B 29/IV	B 29/IV
U:	230	230	230	190	210	190

i INF 2	5,6	49	50	55,56	58	60	61
→ i INF 3	11	35	29	29	29	29	29
→ 2/I	B 11/I	C 35/I	C 29/I	C 29/I	C 29/I	C 29/I	C 29/I
→ 2/II	B 11/II	C 35/II	C 29/II	C 29/II	C 29/II	C 29/II	C 29/II
→ 2/III	B 11/III	C 35/III	C 29/III	C 29/III	C 29/III	C 29/III	C 29/III
→ B/IV	=	B 35/IV	B 29/IV	B 29/IV	B 29/IV	B 29/IV	B 29/IV
U:	230	190	190	210	190	190	190

А/11 -ий, -ий

i INF 2	50-52	53-56	58
→ iINF 3	25	25	25
→ 2/I	С 25/I	С 25/I	С 25/I
→ 2/II	С 25/II	С 25/II	С 25/II
→ 2/III	С 25/III	С 25/III	С 25/III
→ В/IV	В 25/IV	В 25/IV	В 25/IV
U:	190	210	190

А/12 -й

i INF 2	1	2	8,10	28-47	59
→ iINF 3	25	1	21	15	25
→ 2/I	В 25/I	-	В 21/I	-	С 25/I
→ 2/II	В 55/II	-	В 21/II	-	С 25/II
→ 2/III	В 25/III	В 1/III	В 21/III	-	С 25/III
→ В/IV	-	-	-	-	В 25/IV
U:	230	230	230	240	190



A/13 -OMY, -EMY

i INF 2	49-52	53-56	58-60	61
→ i INF 3	27	27	27	27
→ 2/I	C 27/I	C 27/I	C 27/I	C 27/I
→ 2/II	C 27/II	C 27/II	C 27/II	C 27/II
→ 2/III	C 27/III	C 27/III	C 27/III	C 27/III
→ B/IV	B 27/IV	B 27/IV	B 27/IV	B 27/IV
U:	190	210	190	190

A/15 -Y10,

A/17 -1010

i INF 2	49-52	53-56	58
→ i INF 3	8	8	8
→ 2/I	-	C 8/I	C 8/I
→ 2/II	-	C 8/II	C 8/II
→ 2/III	-	C 8/III	C 8/III
→ B/IV	-	B 8/IV	B 8/IV
U:	190	210	190

A/14 - Y

i INF 2	1,2,11,14,15	5,6	9,12	13	28-47	60	61	62	63
→ i INF 3	4	8	16	26	2 + sk	8	8	KII	KII
→ 2/I	B 4/I	B 8/I	B 16/I	B 26/I	-	C 8/I	C 8/I	-	-
→ 2/II	B 4/II	B 8/II	B 16/II	B 26/II	-	C 8/II	C 8/II	-	-
→ 2/III	B 4/III	B 8/III	B 16/III	B 26/III	-	C 8/III	C 8/III	-	-
→ B/IV	-	-	-	-	-	B 8/IV	B 8/IV	-	-
U:	230	230	230	230	240	190	190	240	1001

A/16 -b10

i INF 2	7	52
→ i INF 3	11	8
→ 2/I	B 11/I	C 8/I
→ 2/II	B 11/II	C 8/II
→ 2/III	B 11/III	C 8/III
→ B/IV	-	B 8/IV
U:	230	190

A/18 -10

i INF 2	1,2,3,14,15	5,6,8	9,10	59	28-47
→ i INF 3	4	8	16	8	2 + sk
→ 2/I	B 4/I	B 8/I	B 16/I	C 8/I	-
→ 2/II	B 4/II	B 8/II	B 16/II	C 8/II	-
→ 2/III	B 4/III	B 8/III	B 16/III	C 8/III	-
→ B/IV	-	-	-	B 8/IV	-
U:	230	230	230	190	240

A/19 -EE

i INF 2	49-50	51	53-54
→ i INF 3	34	30	30
→ 2/I	-	C 30/I	C 30/I
→ 2/II	-	C 30/II	C 30/II
→ 2/III	-	C 30/III	C 30/III
→ B/IV	-	B 30/IV	B 30/IV
U:	1005	190	210

A/20 - OE

i INF 2	49-50	55-56	58
→ i INF 3	30	30	30
→ 2/I	C 30/I	C 30/I	C 30/I
→ 2/II	C 30/II	C 30/II	C 30/II
→ 2/III	C 30/III	C 30/III	C 30/III
→ B/IV	B 30/IV	B 30/IV	B 30/IV
U:	1005	210	190

INF 2	49-51	53-56	58
iINF 3	32	32	32
2/I	C32/I	C32/I	C32/I
2/II	C32/II	C32/II	C32/II
2/III	C32/III	C32/III	C32/III
B/IV	B32/IV	B32/IV	B32/IV
U:	190	210	190

i INF 2	1,2,11,14,15	12	5,6	9	10	60	49-50	62	63
→ iINF 3	6	18	28	31	30	32	34	KII	KII
→ 2/I	B 6/I	B18/I	B28/I	B31/I	B30/I	C32/I	-	-	-
→ 2/II	B 6/II	B18/II	B28/II	B31/II	B30/II	C32/II	-	-	-
→ 2/III	-	-	B28/III	B31/III	B30/III	C32/III	-	-	-
→ B/IV	-	-	-	-	-	B32/IV	-	-	-
U:	230	230	230	230	230	190	240	240	1001

А/24 -АМИ, -ЯМИ

i INF 2	1 - 17
→ i INF 3	23
→ 2/I	B 23/I
→ 2/II	B 23/II
→ 2/III	B 23/III
→ B/IV	-
U:	230

А/25- -ИМИ, -ИМИ

i INF 2	49 - 51	53 - 56	58 - 60	61
→ i INF 3	23	23	23	23
→ 2/I	B 23/I	B 23/I	B 23/I	B 23/I
→ 2/II	B 23/II	B 23/II	B 23/II	B 23/II
→ 2/III	B 23/III	B 23/III	B 23/III	B 23/III
→ B/IV	B 23/IV	B 23/IV	B 23/IV	B 23/IV
U:	190	210	190	190

i INF 2	49 - 51	53 - 56	58 - 60	61
→ i INF 2	26	26	26	26
→ 2/I	C 26/I	C 26/I	C 26/I	C 26/I
→ 2/II	C 26/II	C 26/II	C 26/II	C 26/II
→ 2/III	C 26/III	C 26/III	C 26/III	C 26/III
→ B/IV	B 26/IV	B 26/IV	B 26/IV	B 26/IV
U:	190	210	190	190

i INF 2	1 - 17
→ i INF 3	24
→ 2/I	B 24/I
→ 2/II	B 24/II
→ 2/III	-
→ B/IV	-
U:	230

1. TIME 2	9.12	<del>28-43, 45-47</del>	44	48	49-51	55	50	61	62	63
← 1 TIME 3	30	13	13-sk	42	42	42	30	30	4-sk	10
← 2/I	B 30/I	-	-	-	-	-	B 30/I	G 30/I	-	-
← 2/II	B 30/II	-	-	-	-	-	G 30/II	G 30/II	-	-
← 2/III	B 30/III	-	-	-	-	-	G 30/III	G 30/III	-	-
← B/IV	-	-	-	-	-	-	B 30/IV	B 30/IV	-	-
U:	230	240	240	240	1005	1005	190	190	240	1001



A/29 - БХ, -ИХ

i INF 3	49-51	53-56	58-60	60
→ i INF 3	31	31	31	31
→ 2/I	С 31/I	С 31/I	С 31/I	С 31/I
→ 2/II	С 31/II	С 31/II	С 31/II	С 31/II
→ 2/III	С 31/III	С 31/III	С 31/III	С 31/III
→ B/IV	В 31/IV	В 31/IV	В 31/IV	В 31/IV
U:	190	210	190	190

A/30 - АЯ, -ЯЯ

i INF 2	49-51	53-56	58
→ i INF 3	7	7	7
→ 2/I	-	-	-
→ 2/II	-	-	-
→ 2/III	С 7/III	С 7/III	С 7/III
→ B/IV	В 7/IV	В 7/IV	В 7/IV
U:	190	210	190

A/31 - A

i INF 2	1,3	5,8	2	9,10	28-47	62	63
→ i INF 3	3	7	27	34	56	KII	KII
→ 2/I	B 3/I	-	B 27/I	B 34/I	-	-	-
→ 2/II	B 3/II	-	B 27/II	B 34/II	-	-	-
→ 2/III	B 3/III	B 7/III	B 27/III	B 34/III	-	-	-
→ B/IV	-	-	-	-	-	-	-
U:	230	230	230	230	240	240	1001

A/33 - Tb

i INF 2	28-47
→ i INF 3	1 + sk
→ 2/I	-
→ 2/II	-
→ 2/III	-
→ B/IV	-
U:	240

i INF 2	1,13	2	5	48	49-51	55	62	63
→ i INF 3	32	19	34	43	43	43	KII	KII
→ 2/I	B 32/I	-	B 34/I	-	-	-	-	-
→ 2/II	B 32/II	-	B 34/II	-	-	-	-	-
→ 2/III	B 32/III	B 19/III	B 34/III	-	-	-	-	-
→ B/IV	-	-	-	-	-	-	-	-
U:	230	230	230	240	1005	1005	240	1001

i INF 2	28-47
→ i INF 3	5+sk
→ 2/I	-
→ 2/II	-
→ 2/III	-
→ B/IV	-
U:	240

A/36-b

i INF 2	1	2	5	7	28-47
→ i INF 3	25	1	21	40	15
→ 2/I	B 25/I	-	B 21/I	B 40/I	-
→ 2/II	B 25/II	-	B 21/II	B 40/II	-
→ 2/III	B 25/III	B 1/III	B 21/III	B 40/III	-
→ B/IV	-	-	-	-	-
U:	230	230	230	230	240

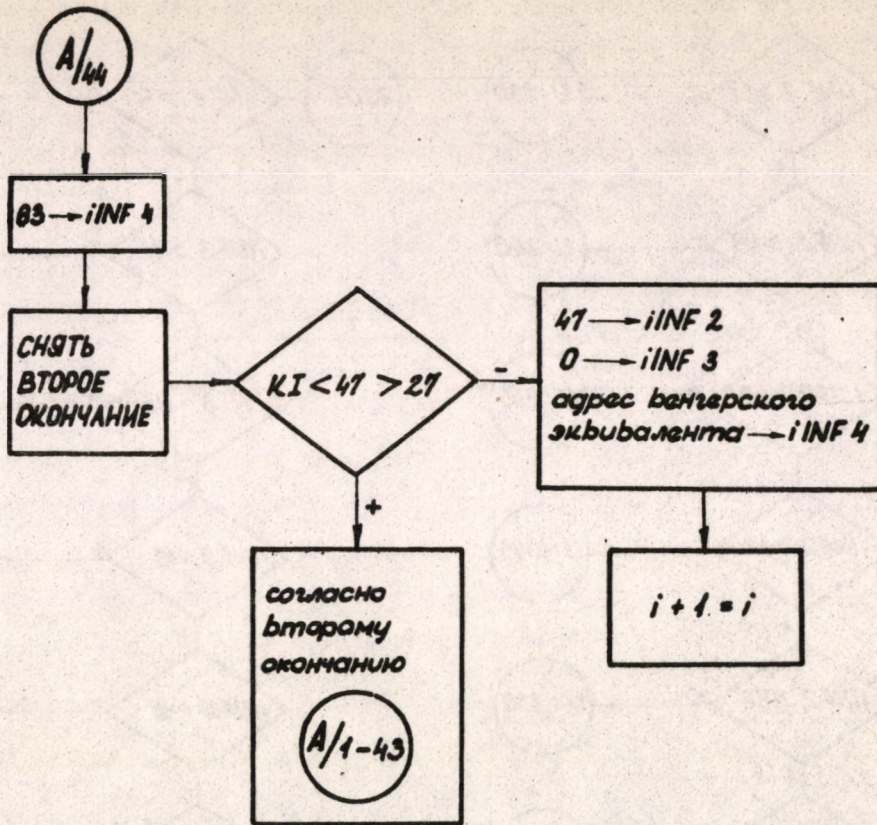
A/38 - ET, -UT

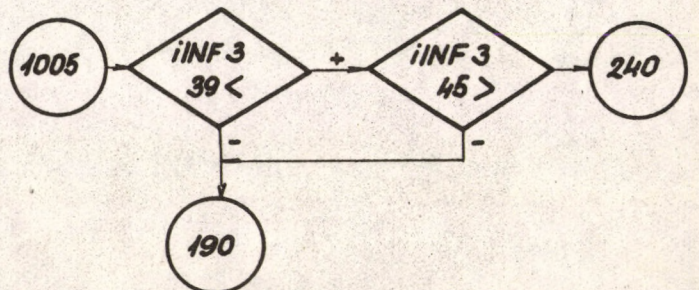
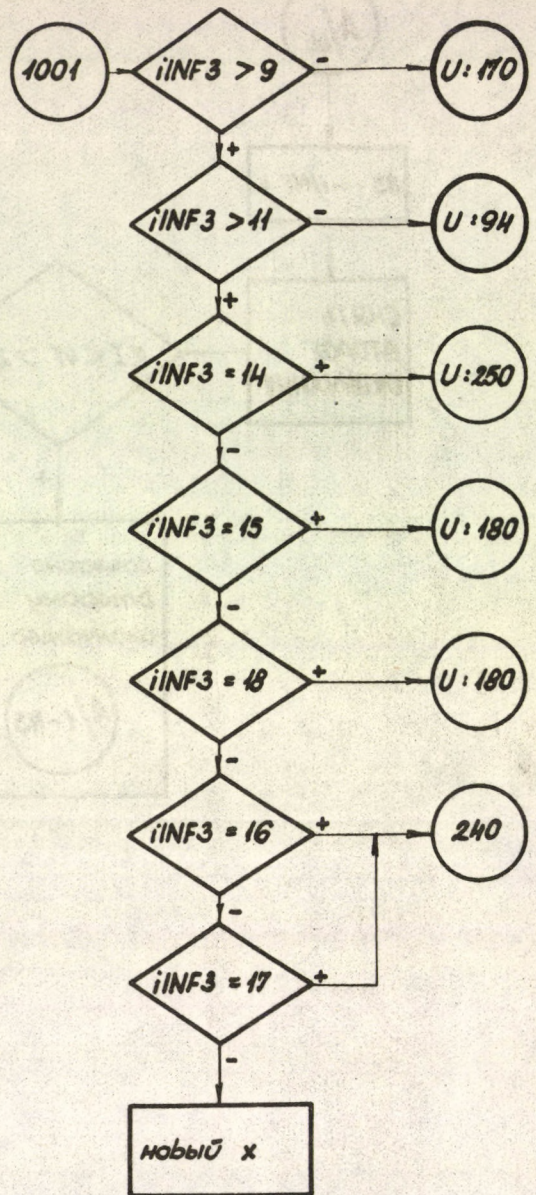
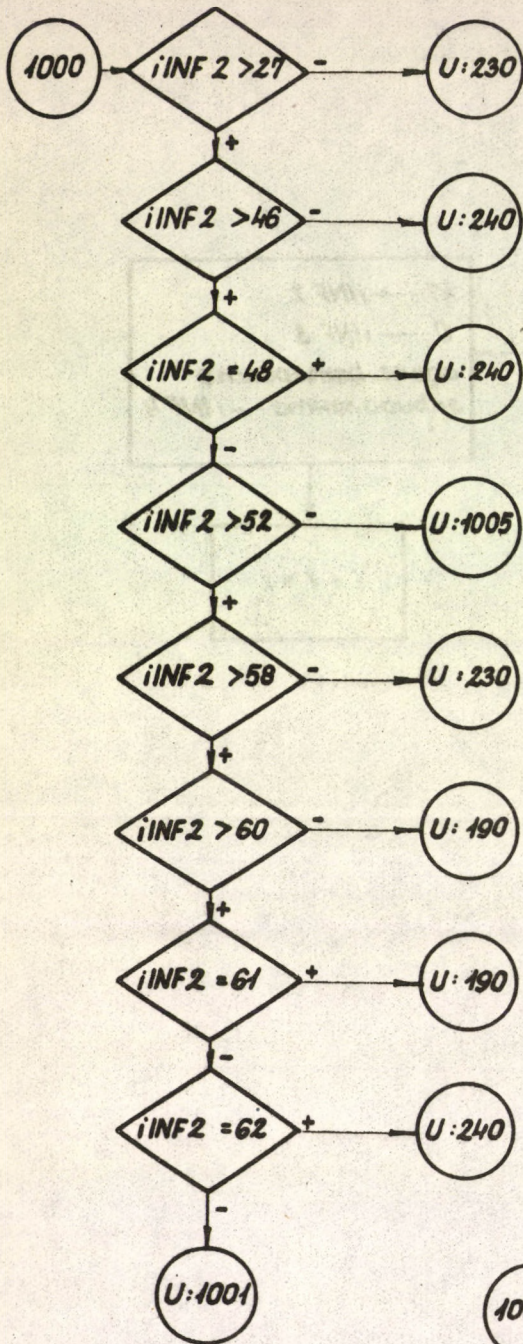
i INF 2	28-47
→ i INF 3	3+sk
→ 2/I	-
→ 2/II	-
→ 2/III	-
→ B/IV	-
U:	240

i INF 2	1	2	3	5	7,8,16,17	10	49-51	59,60	61	62	63
→ i INF 3	32	19	35	36	37	18	43	32	32	KII	KII
→ 2/I	B 32/I	-	B 35/I	B 36/I	B 37/I	B 18/I	-	C 32/I	C 32/I	-	-
→ 2/II	B 32/II	-	B 35/II	B 36/II	B 37/II	B 18/II	-	C 32/II	C 32/II	-	-
→ 2/III	B 32/III	B 19/III	B 35/III	B 36/III	B 37/III	-	-	C 32/III	C 32/III	-	-
→ B/IV	-	-	-	-	-	-	-	B 32/IV	B 32/IV	-	-
U:	230	230	230	230	230	230	240	190	190	240	1001

A/42 - 0

i INF 2	2,5	1,11,13,14	5,9,17	28-43 45-47	44	48	49-51	55
→ i INF 3	1	25	21	11	11+sk	40	40	40
→ 2/I	-	B 25/I	B 21/I	-	-	-	-	-
→ 2/II	-	B 25/II	B 21/II	-	-	-	-	-
→ 2/III	B 1/III	B 25/III	B 21/III	-	-	-	-	-
→ B/IV	-	-	-	-	-	-	-	-
U:	230	230	230	240	240	240	240	240







	B 1/III	B 3/I-II-III	B 4/I-II-III	B 5/I-II-III	B 6/I-II	B 7/III	B 8/I-II-III	B 11/I-II-III
S	1	3	4	5	6	7	8	11
A	0,25,35	0,26	0,27	0,33	0,28	0,7	0,8	0,29,35
Nu	0,25	0,26	0,27	0,33	0,28	0,7	0,8	0,29
Pn	0,25,35	0,26	0,27	0,33	0,28	0,7	0,8	0,29,35
Pr	-	0,1,9	0,2,8	0,4,6,9	5,7,8	-	0,3,6,7	0,4,6,9
S	1	3	4	5	6	7	8	11
A	1	3	4	5	6	7	8	11
Nu	1	3	4	5	6	7	8	11
Pn	1	3	4	5	6	7	8	11
Pr	-	1	2	4	5	-	3	4

S	B15/I-II-III	B16/I-II-III	B17/I-II-III	B18/I-II	B19/III	B21/I-II-III	B22/I-II-III	B23/I-II-III	B24/I-II
S	15	16	17	18	19	21	22	23	24
A	0,26	0,27	0,33	0,28	0,32	0,31	0,33	0,23	0,31
Nu	0,26	0,27	0,33	0,28	0,32	0,31	0,33	0,23	0,31
Pn	0,26	0,27	0,33	0,28	0,32	0,31	0,33	0,23	0,31
Pr	0,1,9	0,2,8	0,4,6,9	5,7,8	-	0,1,9	0,2,8	0,4,6,9	5,7,8
S	15	16	17	18	19	21	22	23	24
A	15	16	17	18	19	21	22	23	24
Nu	15	16	17	18	19	21	22	23	24
Pn	15	16	17	18	19	21	22	23	24
Pr	1	2	4	5	-	1	2	4	5

	B 25/II	B 25/II	B 25/III
S	25	25	25
A	0, 25, 35	-	0, 25, 35
Nu	0, 25	-	0, 25
Pn	0, 25, 35	-	0, 25, 35
Pr	3, 6, 7	3, 6, 7	-
S	2	2	25
A	2	-	25
Nu	2	-	25
Pn	2	-	25
Pr	3	3	-

B 26/I.	2	3	4	B 26/II	2	3	B 26/III
1				1			
26	26	26	26	26	26	26	26
0, 27	0, 28	0, 27	0, 28	-	-	-	0, 27
0, 27	0, 28	0, 27	0, 28	-	-	-	0, 27
0, 27	0, 28	0, 27	0, 28	-	-	-	0, 27
8	8	2	5, 7	2	5, 7	8	-
4	6	4	6	4	6	26	4
4	6	4	6	-	-	-	4
4	6	4	6	-	-	-	4
4	6	4	6	-	-	-	4
2	5	2	5	2	5	8	-

	B 27/I.		B 27/II		B 27/III
	1	2	1	2	
S	27	27	27	27	27
A	0,26	0,26	-	-	0,26
Nu	0,26	0,26	-	-	0,26
Pn	0,26	0,26	-	-	0,26
Pr	3,6,7	1,9	3,6,7	1,9	-
S	2	3	2	3	27
A	2	3	-	-	0,26
Nu	2	3	-	-	0,26
Pn	2	3	-	-	0,26
Pr	3	1	3	1	-

	B 28/I		B 28/II		B 28/III
	1	2	1	2	
S	28	28	28	28	28
A	0,29,35	<del>0,29,35</del>	-	-	0,29,35
Nu	0,29	0,29	-	-	0,29
Pn	0,29,35	0,29,35	-	-	0,29,35
Pr	2,8	5,7	2,8	5,7	-
S	10	12	10	12	10
A	10	12	-	-	10
Nu	10	12	-	-	10
Pn	10	12	-	-	10
Pr	2	5	2	5	-

	B 30/I	B 30/II	B 30/III
S	30	30	30
A	0,30	-	0,30
Nu	0,30	-	0,30
Pn	0,30	-	0,30
Pr	3,6,7	3,6,7	-
S	14	14	30
A	14	-	30
Nu	14	-	30
Pn	14	-	30
Pr	3	3	-

B 31/I		B 31/II			B 31/III
1	2	1	2	3	
31	31	31	31	31	31
0,28	0,30	-	-	-	0,30
0,28	0,30	-	-	-	0,30
0,28	0,30	-	-	-	0,30
5,7,8	3,6,7	5,8	3,6	7	-
18	14	18	14	42	31
18	14	-	-	-	30
18	14	-	-	-	30
18	14	-	-	-	30
5	3	5	3	7	-

	B 32/I	B 32/II	B 32/III
S	32	32	32
A	0,32	-	0,32
Nu	0,32	-	0,32
Pn	0,32	-	0,32
Pr	3,6,7	3,6,7	-
S	20	20	32
A	20	-	32
Nu	20	-	32
Pn	20	-	32
Pr	3	3	-

B 33/I		B 33/II		B 33/III
1	2	1	2	
33	33	33	33	33
0,31	0,31	-	-	0,31
0,31	0,31	-	-	0,31
0,31	0,31	-	-	0,31
3,6,7	1,9	3,6,7	1,9	-
20	21	20	21	33
20	21	-	-	31
20	21	-	-	31
20	21	-	-	31
3	1	3	1	-

	B 34/I		B 34/II		B 34/III	
	1	2	1	2	1	2
S	34	34	34	34	34	34
A	0,32	0,26	-	-	0,26	0,32
Nu	0,32	0,26	-	-	0,26	0,32
Pn	0,32	0,26	-	-	0,26	0,32
Pr	3,6,7	1,9	3,6,7	1,9	-	-
S	20	15	20	15	15	32
A	20	15	-	-	15	32
Nu	20	15	-	-	15	32
Pn	20	15	-	-	15	32
Pr	3	1	3	1	-	-

	B 35/I		B 35/II		B 35/III
	1	2	1	2	
S	35	35	35	35	35
A	0,32	0,23	-	-	0,32
Nu	0,32	0,28	-	-	0,32
Pn	0,32	0,28	-	-	0,32
Pr	3,6	5,7,8	3,6	5,7,8	-
S	20	6	20	6	32
A	20	6	-	-	32
Nu	20	6	-	-	32
Pn	20	6	-	-	32
Pr	3	5	3	5	-

	B 36/I		B 36/II		B 36/III	
	1	2	1	2	1	2
S	36	36	36	36	36	36
A	0, 29, 35	0, 32	-	-	0, 29, 35	0, 32
Na	0, 29	0, 32	-	-	0, 29	0, 32
Pn	0, 29, 35	0, 32	-	-	0, 29, 35	0, 32
Pr	1, 9	3, 6, 7	1, 9	3, 6, 7	-	-
S	9	20	9	20	9	32
A	9	20	-	-	9	32
Na	9	20	-	-	9	32
Pn	9	20	-	-	9	32
Pr	1	3	1	3	-	-

	B 37/I				
	1	2	3	4	5
S	37	37	37	37	37
A	0, 32	0, 29, 35	0, 29, 35	0, 29, 35	0, 29, 35
Na	0, 32	0, 29	0, 29	0, 29	0, 29
Pn	0, 32	0, 29, 35	0, 29, 35	0, 29, 35	0, 29, 35
Pr	3, 6, 7	5, 7	1, 9	2	8
S	20	12	9	10	28
A	20	12	9	10	29
Na	20	12	9	10	29
Pn	20	12	9	10	29
Pr	3	5	1	2	8



	B 37/II						B 37/III	
	1	2	3	4	5	6	1	2
S	37	37	37	37	37	37	37	37
A	-	-	-	-	-	-	0,29,35	0,32
Nu	-	-	-	-	-	-	0,29,35	0,32
Pn	-	-	-	-	-	-	0,29,35	0,32
Pr	3,6	5	7	1,9	2	8	-	-
S	20	12	43	9	10	28	29	32
A	-	-	-	-	-	-	29	32
Nu	-	-	-	-	-	-	29	32
Pn	-	-	-	-	-	-	29	32
Pr	3	5	7	1	2	8	-	-

	B 38/I		B 38/II		B 38/III		B 39/I		B 39/II		B 39/III	
	1	2	1	2	1	2	1	2	1	2	1	2
S	38	38	38	38	38	38	39	39	39	39	39	39
A	0,26	0,26	-	-	0,26	0,32	0,26	0,32	-	-	0,26	0,32
Nu	0,26	0,26	-	-	0,26	0,32	0,26	0,32	-	-	0,26	0,32
Pn	0,26	0,26	-	-	0,26	0,32	0,26	0,32	-	-	0,26	0,32
Pr	3,6,7	1,9	3,6,7	1,9	-	-	1,9	3,6,7	1,9	3,6,7	-	-
S	2	3	2	3	27	19	3	20	3	20	3	32
A	2	3	-	-	26	19	3	20	-	-	3	32
Nu	2	3	-	-	26	19	3	20	-	-	3	32
Pn	2	3	-	-	26	19	3	20	-	-	3	32
Pr	3	1	3	1	-	-	1	3	1	3	-	-

	B 40/I	B 40/II	B 40/III	
			1	2
S	40	40	40	40
A	0,8	-	0,7	0,8
Na	0,8	-	0,7	0,8
Pn	0,8	-	0,7	0,8
Pr	3,6,7	3,6,7	-	-
S	8	8	7	8
A	8	-	7	8
Na	8	-	7	8
Pn	8	-	7	8
Pr	3	3	-	-

B 41/I	B 41/II	B 41/III	
		1	2
41	41	41	41
0,29	-	0,29	0,32
0,29	-	0,29	0,32
0,29	-	0,29	0,32
1,9	1,9	-	-
9	9	9	19
9	-	9	19
9	-	9	19
9	-	9	19
1	1	-	-

	C 7/III	C 8/I-III	C 23/I-III	C 27/I-III	C 28/I-II
A	0,7	0,8	0,23	0,27	0,28
Nu	0,7	0,8	0,23	0,27	0,28
Pn	0,7	0,8	0,23	0,27	0,28
Pr	-	0,3,6,7	0,4,6,9	0,2,8	-
A	7	8	23	27	28
Nu	7	8	23	27	28
Pn	7	8	23	27	28
Pr	-	3	4	2	5

	C 35/III	C 35/III	C 35/I-II	C 35/I-II	C 35/I-II	C 35/I-II	C 35/I-II	C 35/I-II	C 35/I-II
A	0,35	0,35	0,35	0,35	0,35	0,35	0,35	0,35	0,35
Nu	0,25	0,29	0,25	0,29	0,29	0,29	0,29	0,29	0,29
Pn	0,25,35	0,29	0,25	0,29	0,29	0,29	0,29	0,29	0,29
Pr	-	-	3,6,7	1	2	4,6	5,7	8	9
A	25	29	2	9	10	11	12	37	39
Nu	25	29	2	9	10	11	12	37	39
Pn	25	29	2	9	10	11	12	37	39
Pr	-	-	3	1	2	4	5	8	9

	C 25/II	c 25/II	C 25/III
A	0,25	0,25	0,25
Nu	0,25	-	0,25
Pn	0,25	-	0,25
Pr	3,6,7	3,6,7	-
A	2	2	25
Nu	2	-	25
Pn	2	-	25
Pr	3	3	

C 26/I	2	C 26/III	2	C 26/III
1		1		
0,26	0,26	0,26	0,26	0,26
0,26	0,26	-	-	0,26
0,26	0,26	-	-	0,26
3,6,7	1,9	3,6,7	1,9	-
2	3	2	3	26
2	3	-	-	26
2	3	-	-	26
3	1	3	1	-

	C 29/I.				C 29/II				C 29/III
	1	2	3	4	1	2	3	4	
A	0,29	0,29	0,29	0,29	0,29	0,29	0,29	0,29	0,29
Nu	0,29	0,29	0,29	0,29	-	-	-	-	0,29
Pn	0,29	0,29	0,29	0,29	-	-	-	-	0,29
Pr	1,9	2	7	8	1,9	2	7	8	-
A	9	10	12	37	9	10	12	37	29
Nu	9	10	12	37	-	-	-	-	29
Pn	9	10	12	37	-	-	-	-	29
Pr	1	2	5	8	1	2	5	8	-

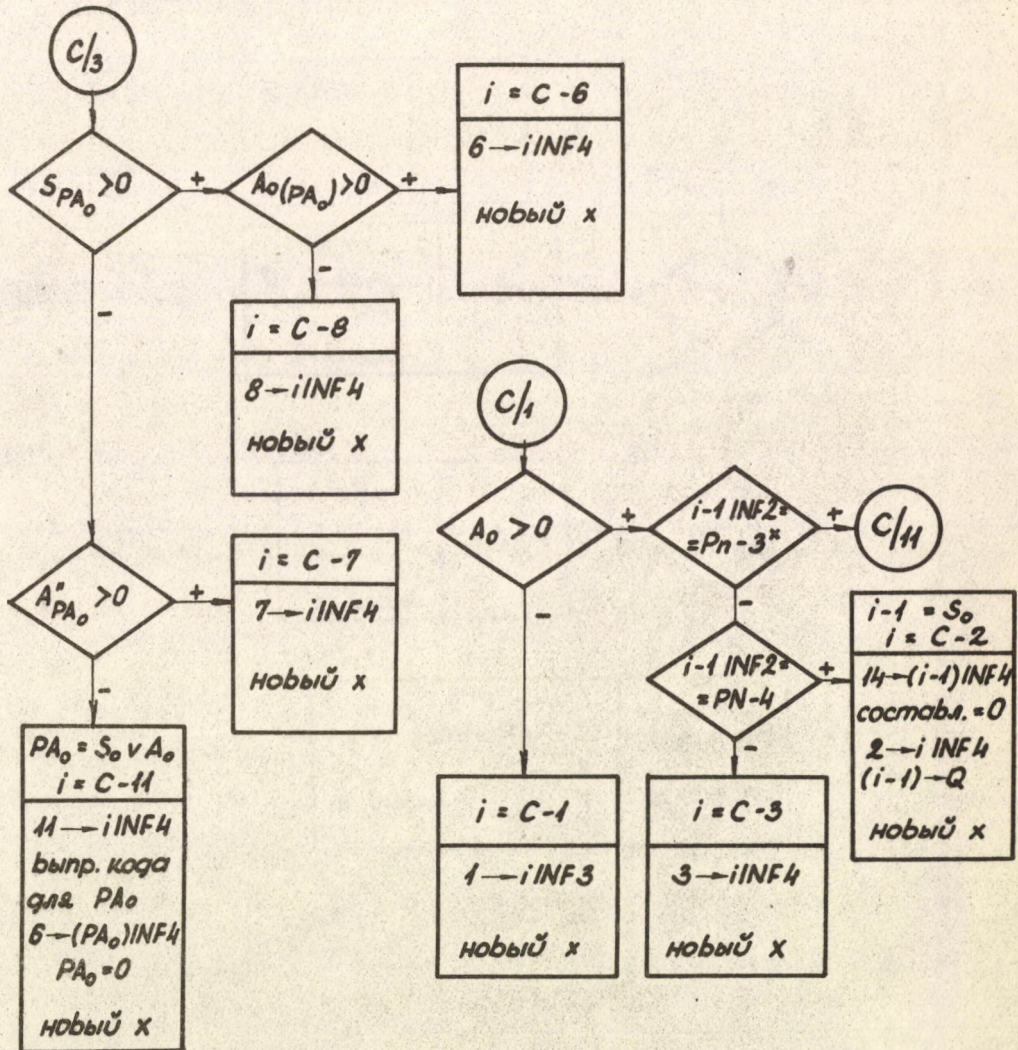
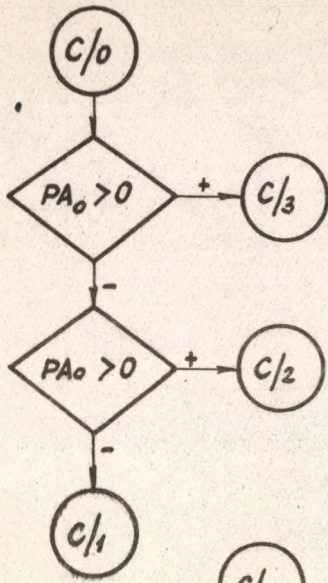
	C30/I	C30/II	C30/III	C31/I 1	2	C31/II 1	2	C31/III	C32/I	C32/II	C32/III
A	0,30	0,30	0,30	0,31	0,31	0,31	0,31	0,31	0,32	0,32	0,32
Nu	0,30	-	0,30	0,31	0,31	-	-	0,31	0,32	-	0,32
Pn	0,30	-	0,30	0,31	0,31	-	-	0,31	0,32	-	0,32
Pr	3,6,7	3,6,7	-	3,6,7	1,9	3,6,7	1,9	-	3,6,7	3,6,7	-
A	14	14	30	20	21	20	21	31	20	20	32
Nu	14	-	30	20	21	-	-	31	20	-	32
Pn	14	-	30	20	21	-	-	31	20	-	32
Pr	3	3	-	3	1	3	1	-	3	3	-

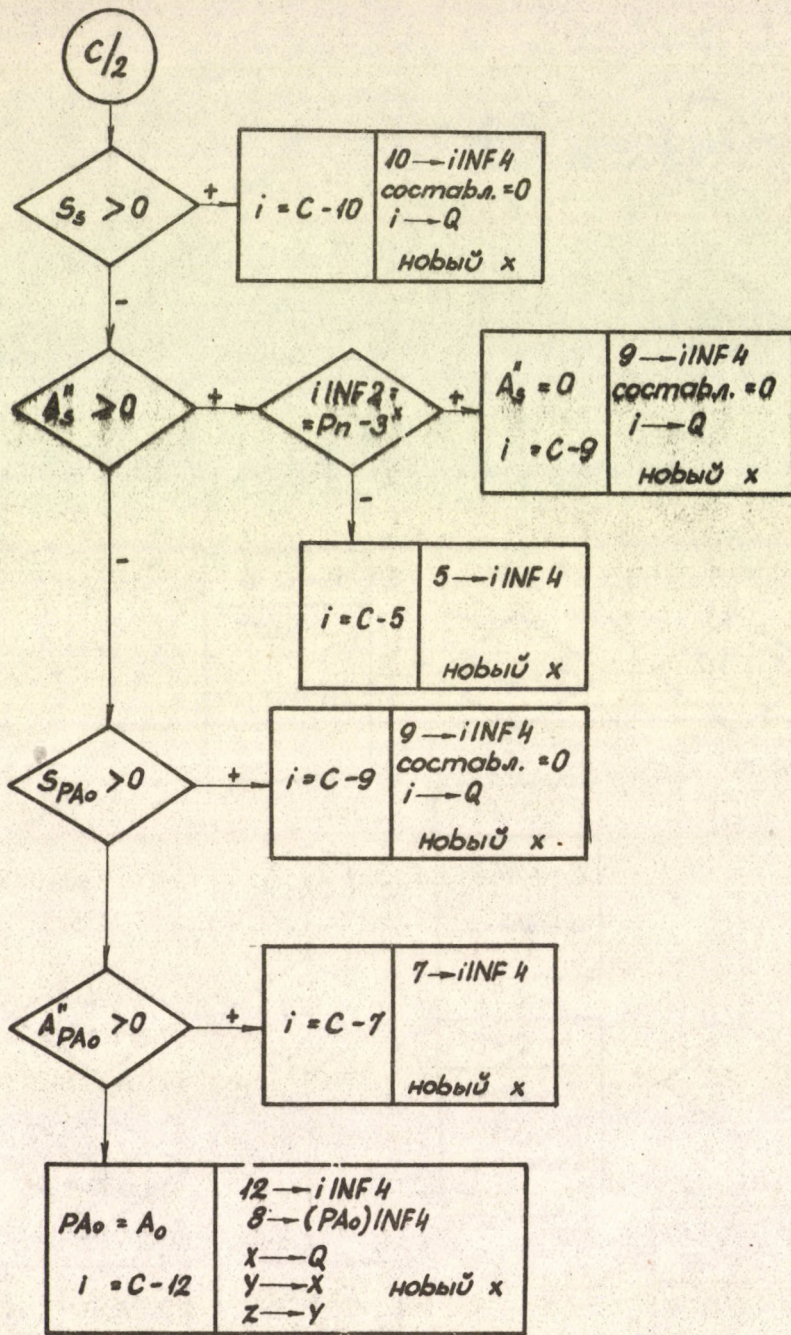


	C 33/I		C 33/II		C 33/III
	1	2	1	2	
A	0,33	0,33	0,33	0,33	0,33
Na	0,33	0,33	-	-	0,33
Pn	0,33	0,33	-	-	0,33
Pr	4,6,9	2,8	4,6,9	2,8	-
A	38	22	38	22	33
Na	38	22	-	-	33
Pn	38	22	-	-	33
Pr	4	2	4	2	-

	B 7/IV	B 8/IV	B 25/IV	B 26/IV	B 27/IV	B 28/IV	B 29/IV
A	7	8	25	26	27	28	29
X } Y } Z }	7	8	2, 25 27, 34	2, 3, 15	4, 16	6, 18	9, 10 11, 12 37

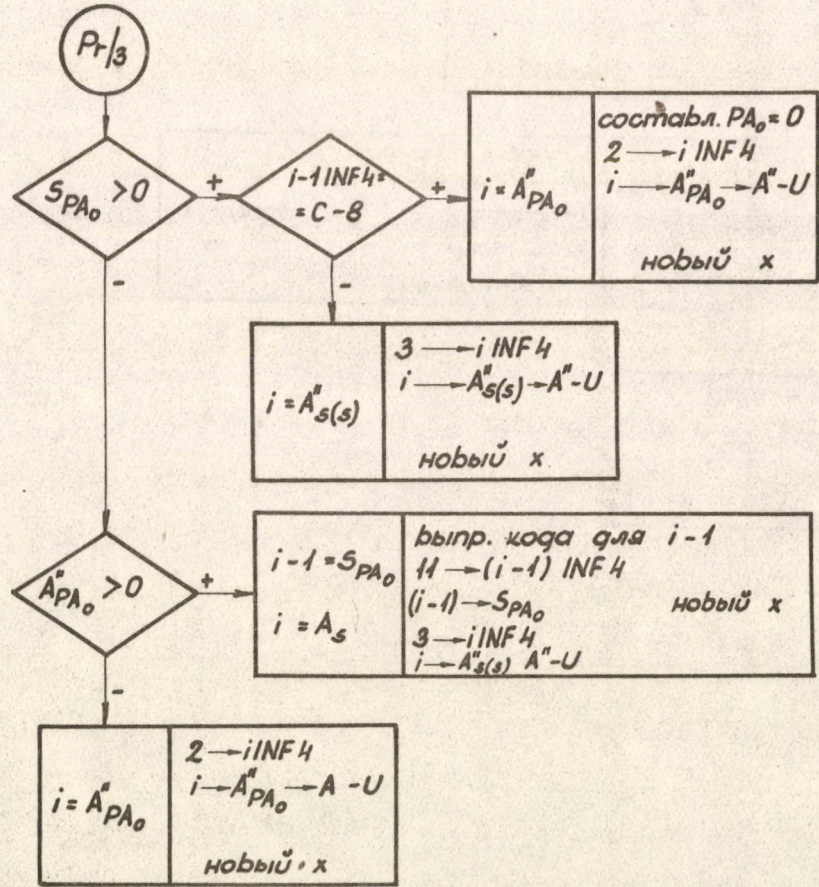
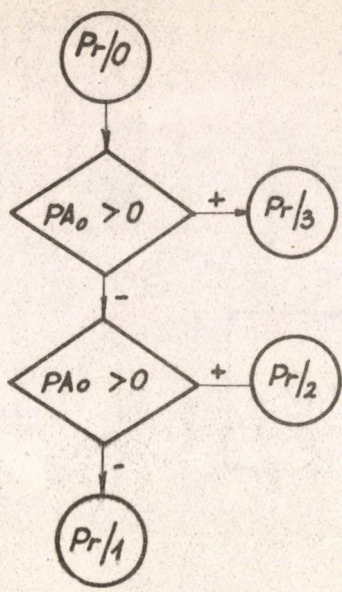
	B 30/IV	B 31/IV	B 32/IV	B 33/IV	B 35/IV	B 23/IV
A	30	31	32	33	35	23
X } Y } Z }	14, 30	20, 21 33	20, 32	5, 17 22	9, 10, 11 12, 25, 28 29	23

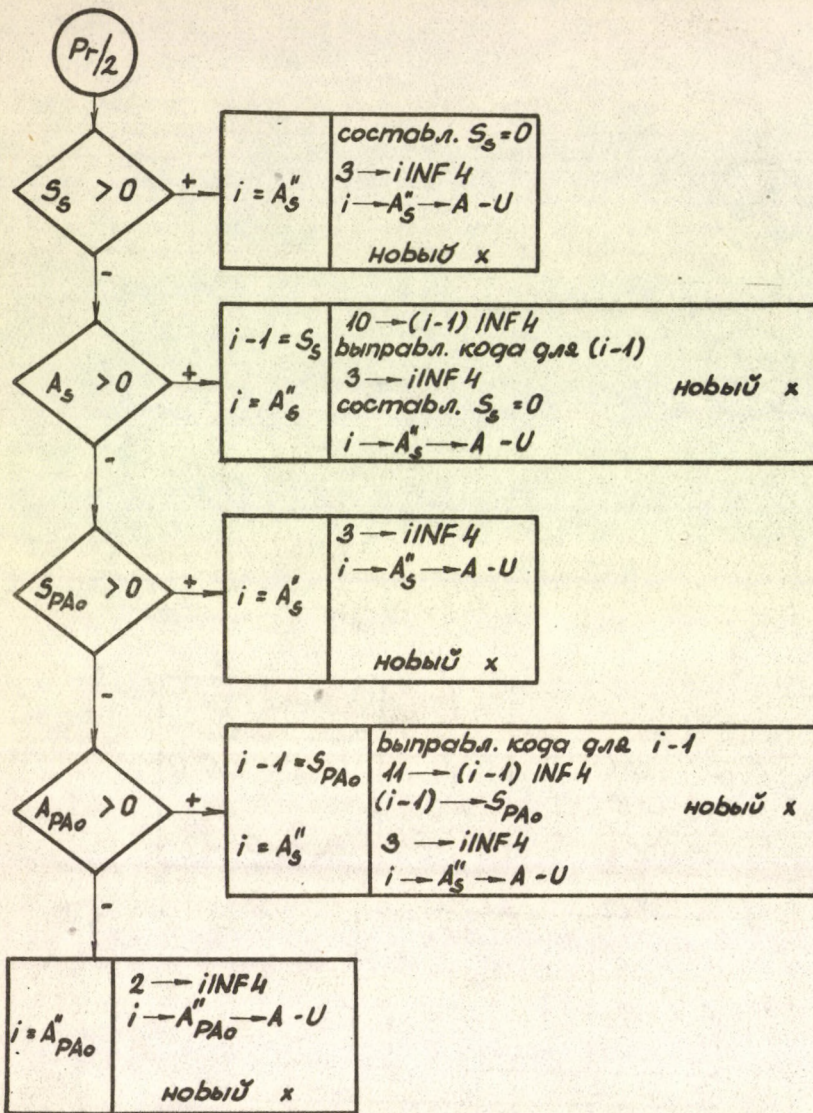


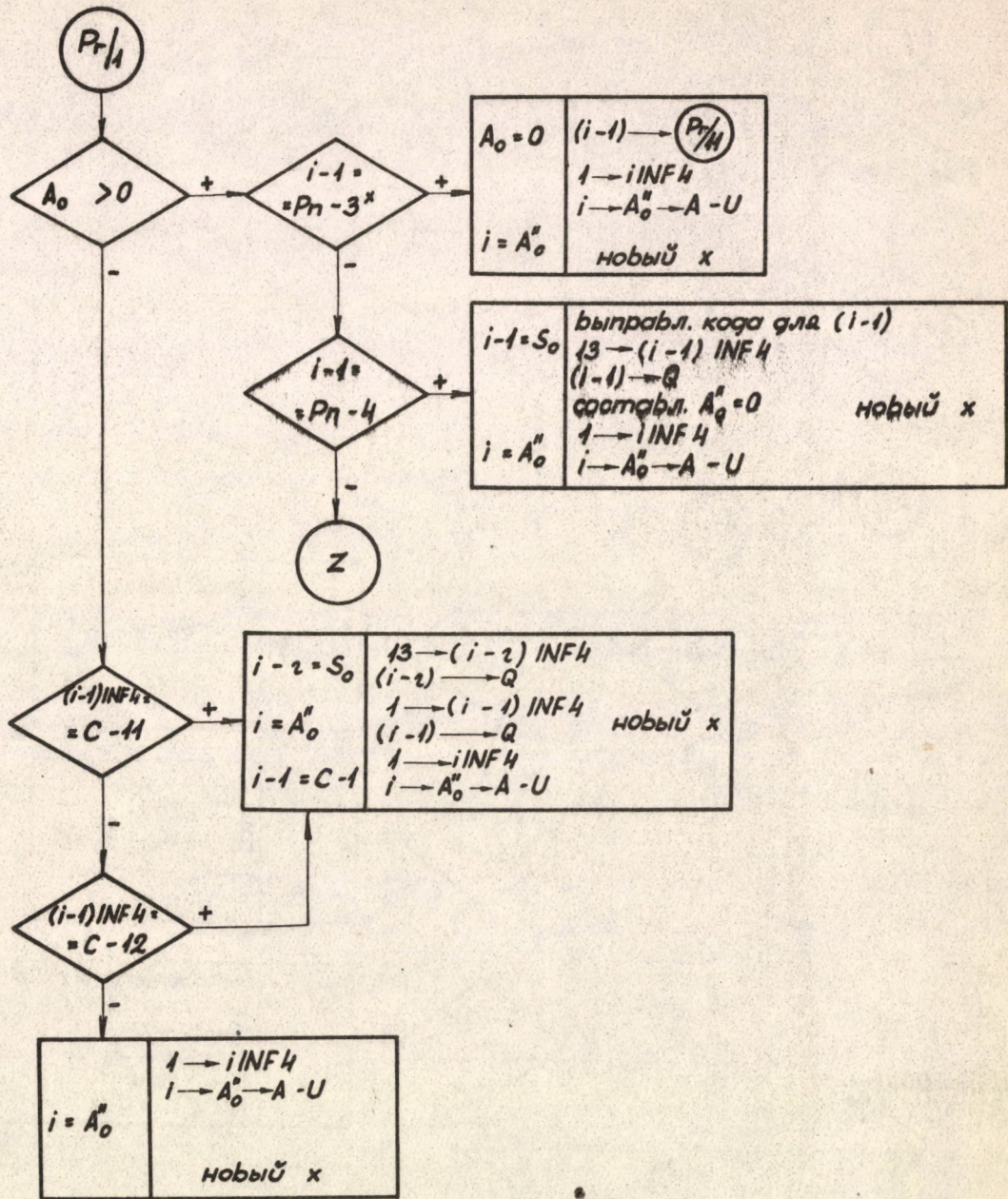


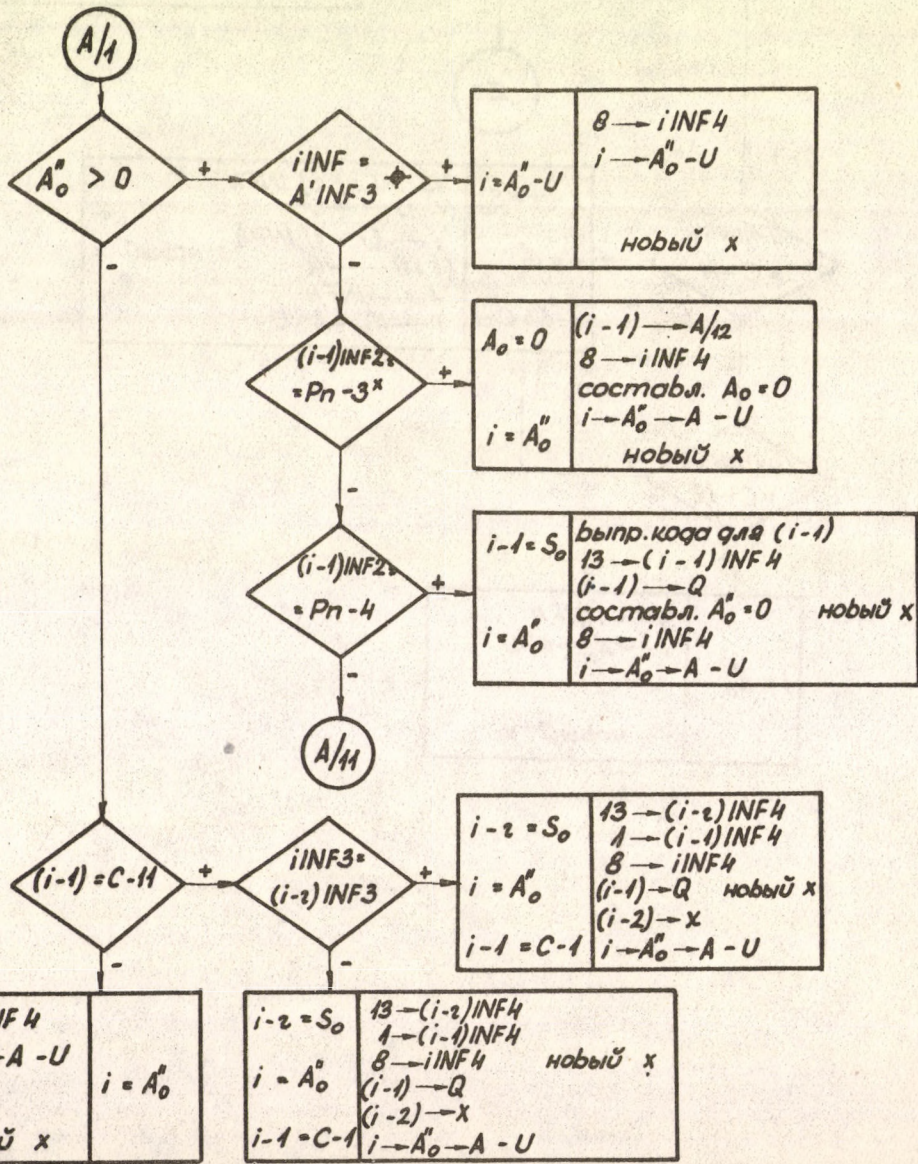
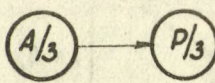
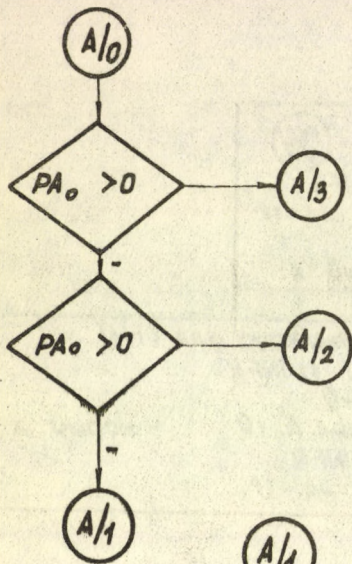
<sup>x</sup> Pn - 3<sup>x</sup> = e10, e2, ux

170 ПРЕДЛОГ









★ по таблицам: В/І, В/ІІ, В/ІІІ

$B \rightarrow i \text{ INF } 4$ $i \rightarrow A_0'' - U$	$i = A_0'' - U$
новый x	

$A_0 = 0$ $(i-1) \rightarrow A/12$ $B \rightarrow i \text{ INF } 4$ составл. $A_0 = 0$ $i \rightarrow A_0'' \rightarrow A - U$	$i = A_0''$
новый x	

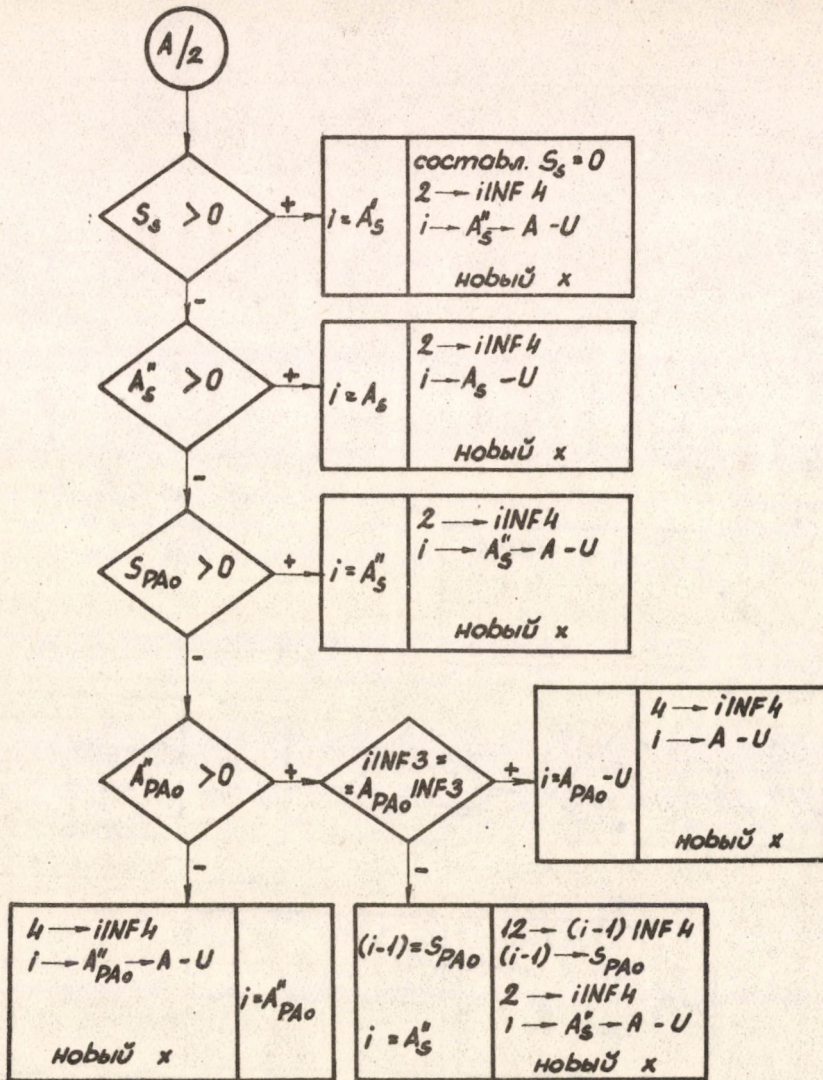
$i-1 = S_0$ $13 \rightarrow (i-1) \text{ INF } 4$ $(i-1) \rightarrow Q$ составл. $A_0'' = 0$	$i = A_0''$
новый x	
$B \rightarrow i \text{ INF } 4$ $i \rightarrow A_0'' \rightarrow A - U$	

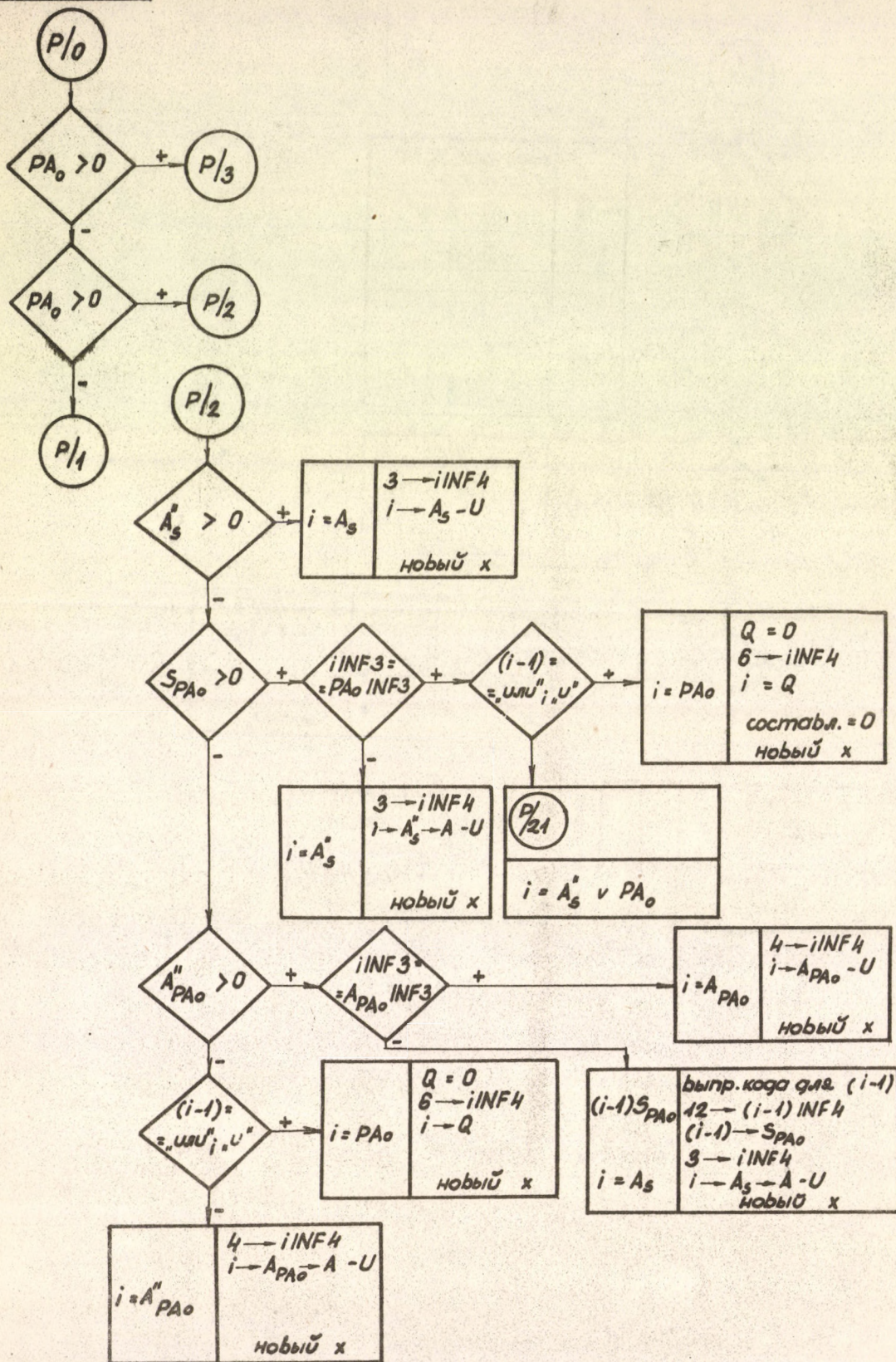
$i-2 = S_0$ $13 \rightarrow (i-2) \text{ INF } 4$ $1 \rightarrow (i-1) \text{ INF } 4$ $B \rightarrow i \text{ INF } 4$ $(i-1) \rightarrow Q$ новый x $(i-2) \rightarrow X$	$i = A_0''$
$i-1 = C-1$ $i \rightarrow A_0'' \rightarrow A - U$	

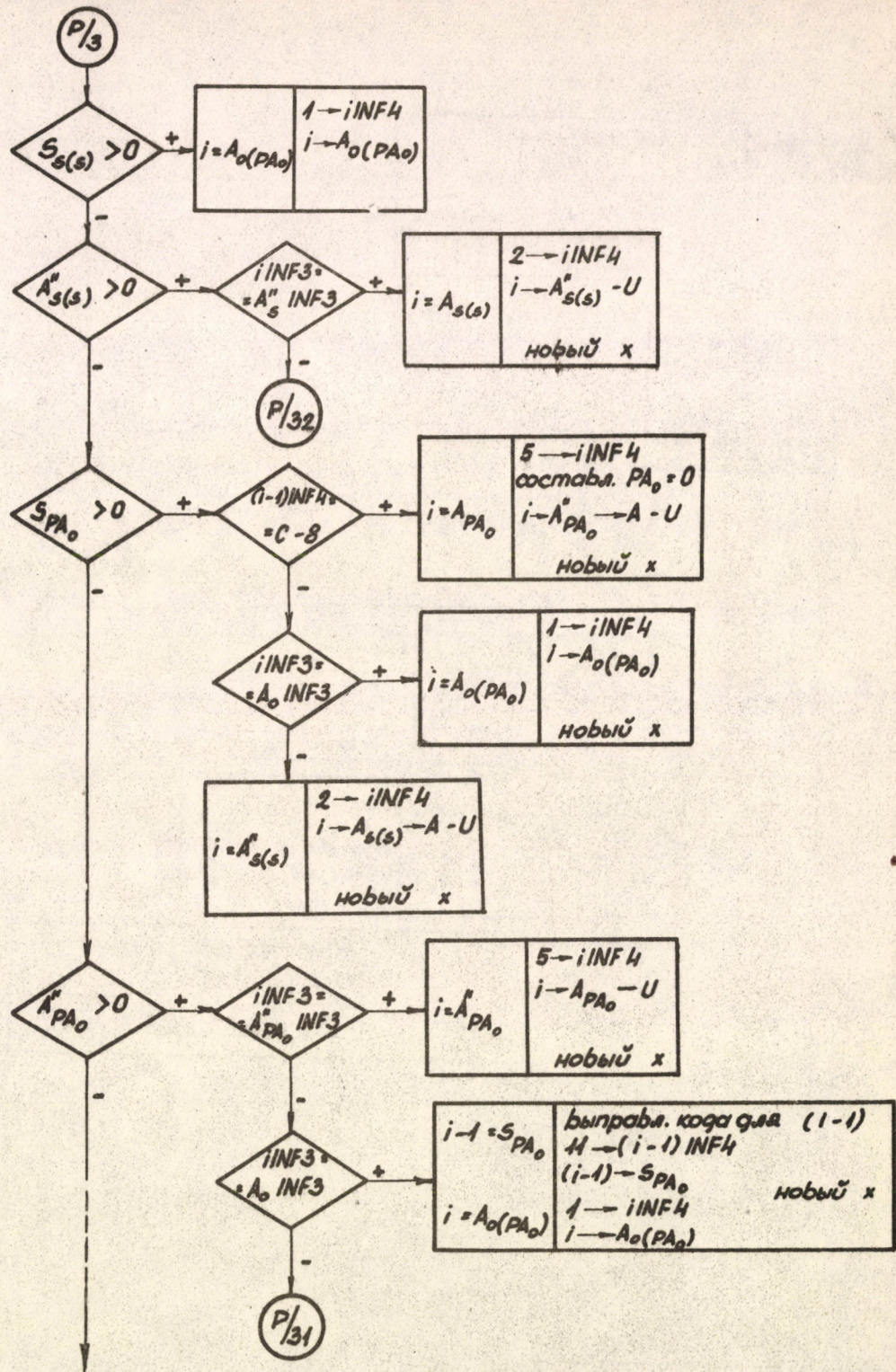
$B \rightarrow i \text{ INF } 4$ $i \rightarrow A_0'' \rightarrow A - U$	$i = A_0''$
новый x	

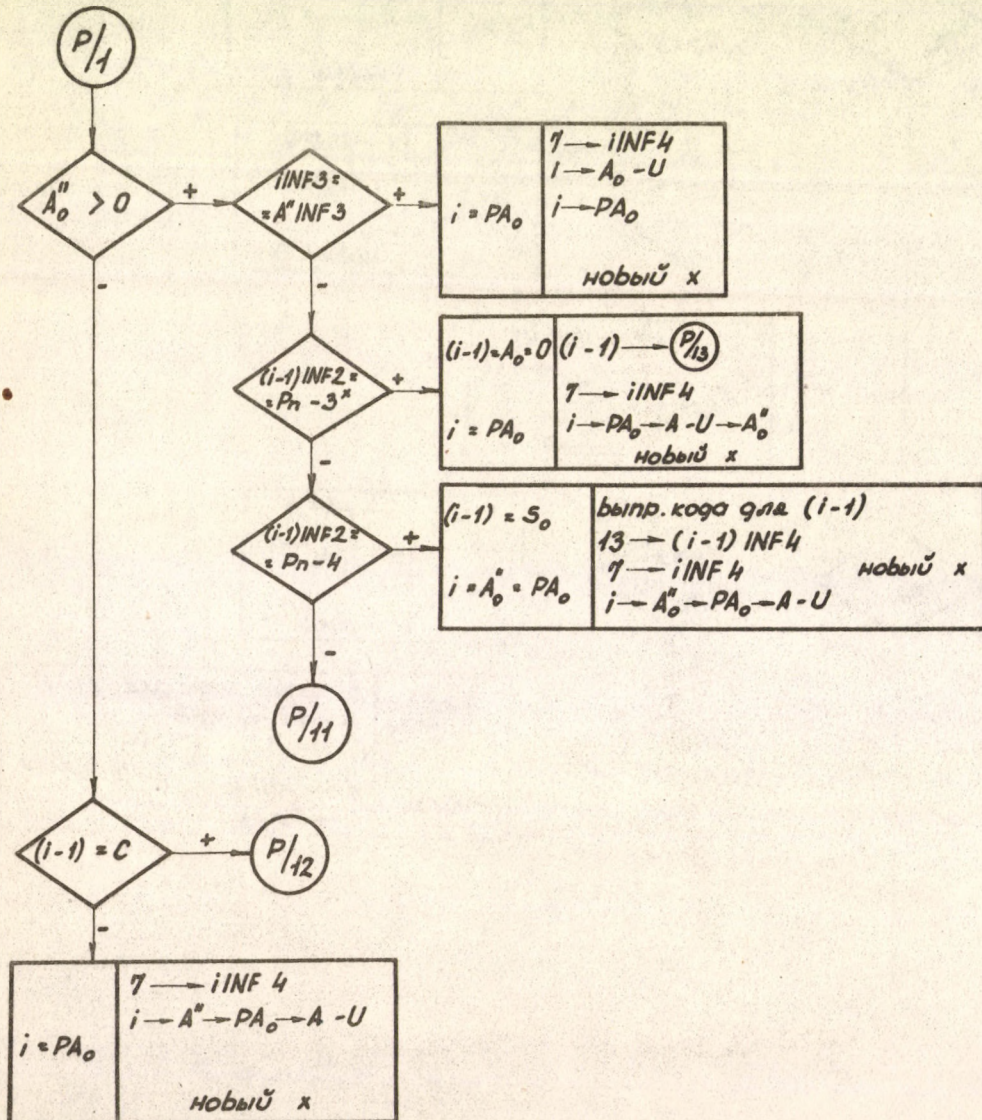
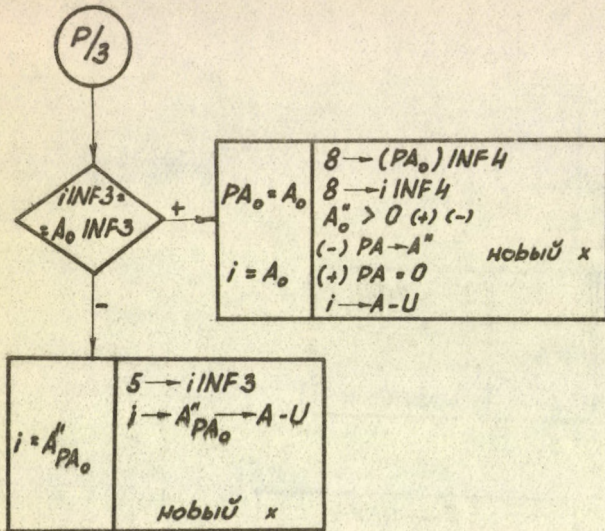
$i-2 = S_0$ $13 \rightarrow (i-2) \text{ INF } 4$ $1 \rightarrow (i-1) \text{ INF } 4$ $B \rightarrow i \text{ INF } 4$ новый x $(i-1) \rightarrow Q$ $(i-2) \rightarrow X$	$i = A_0''$
$i-1 = C-1$ $i \rightarrow A_0'' \rightarrow A - U$	

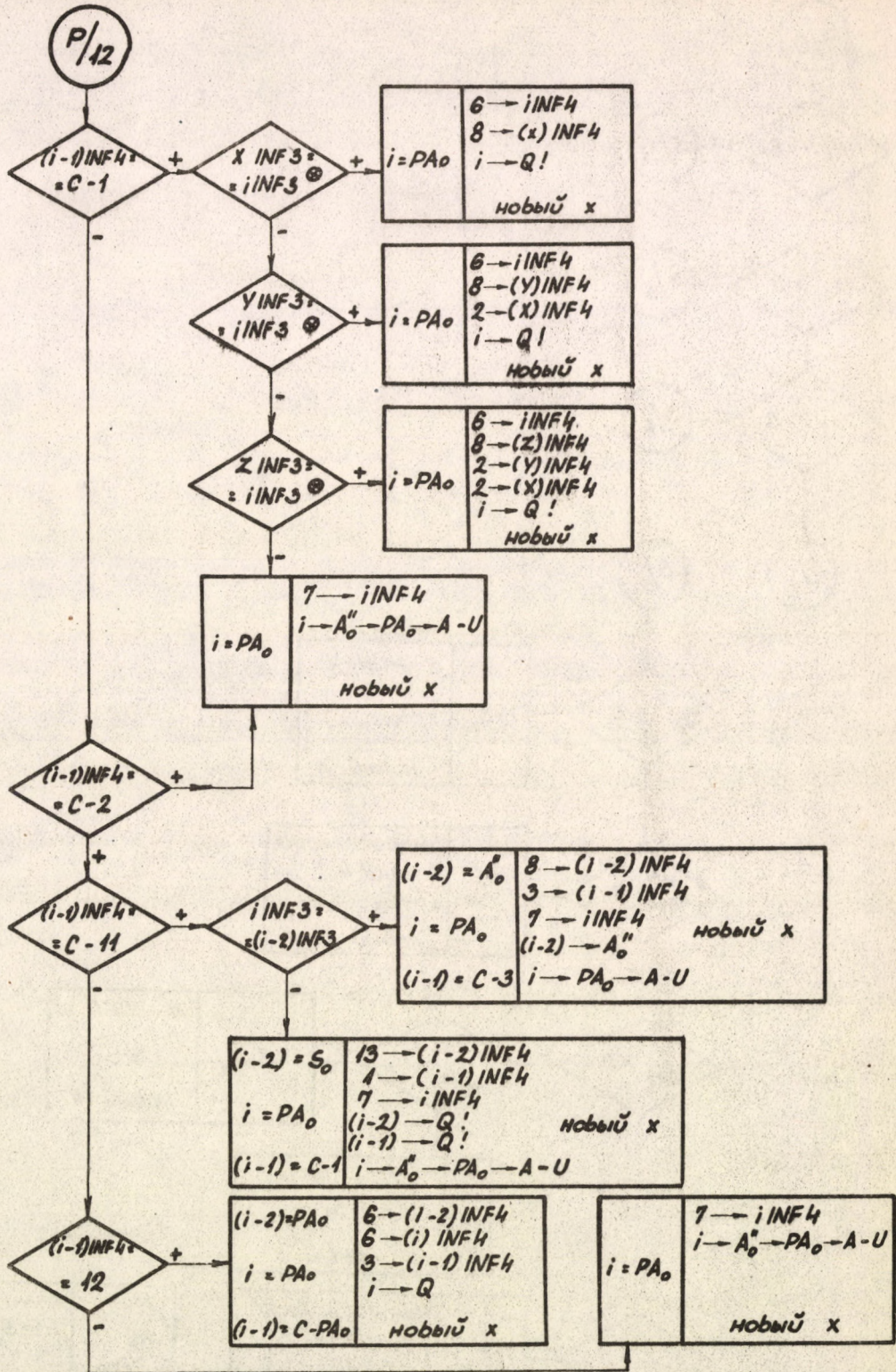




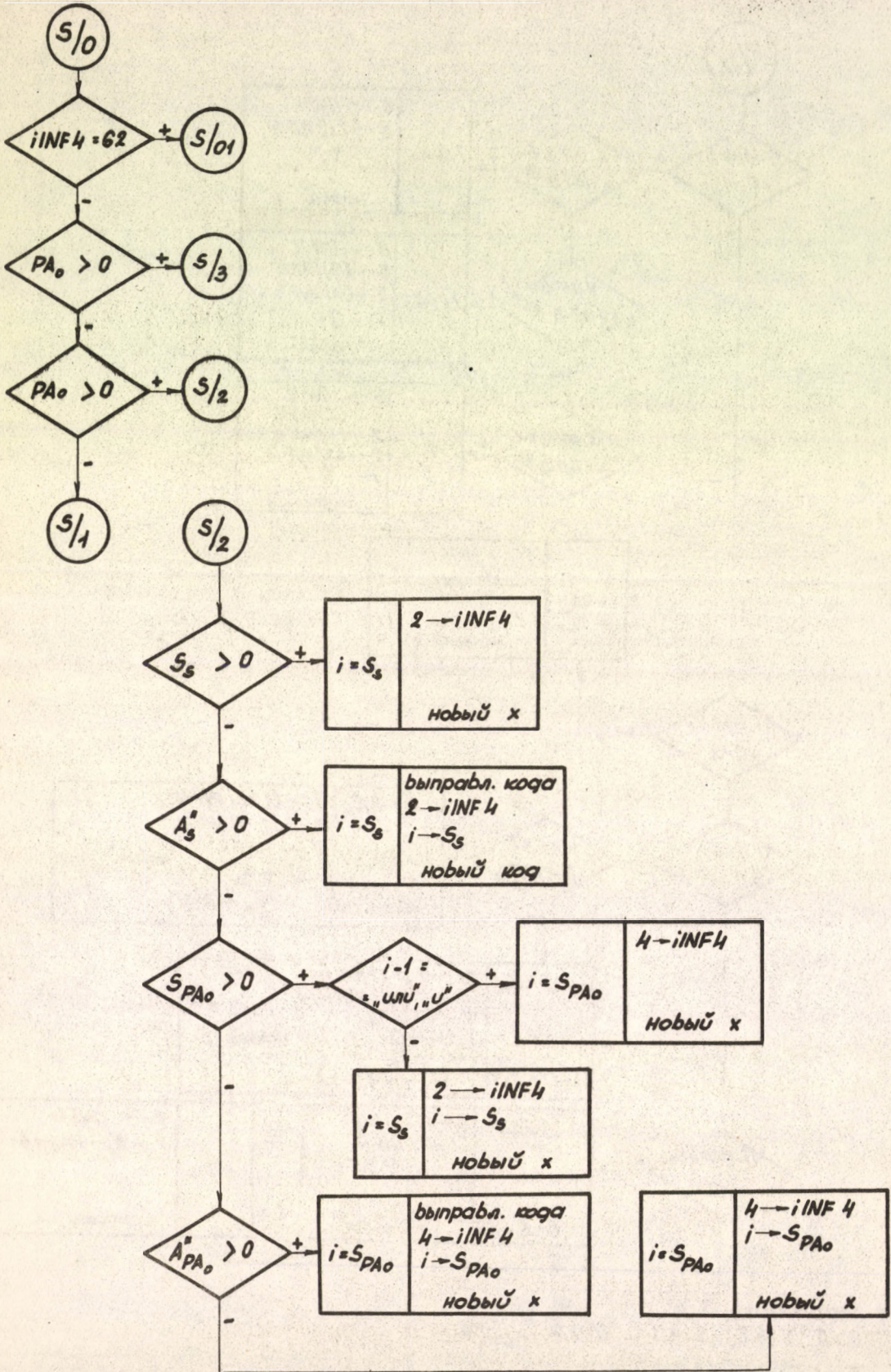


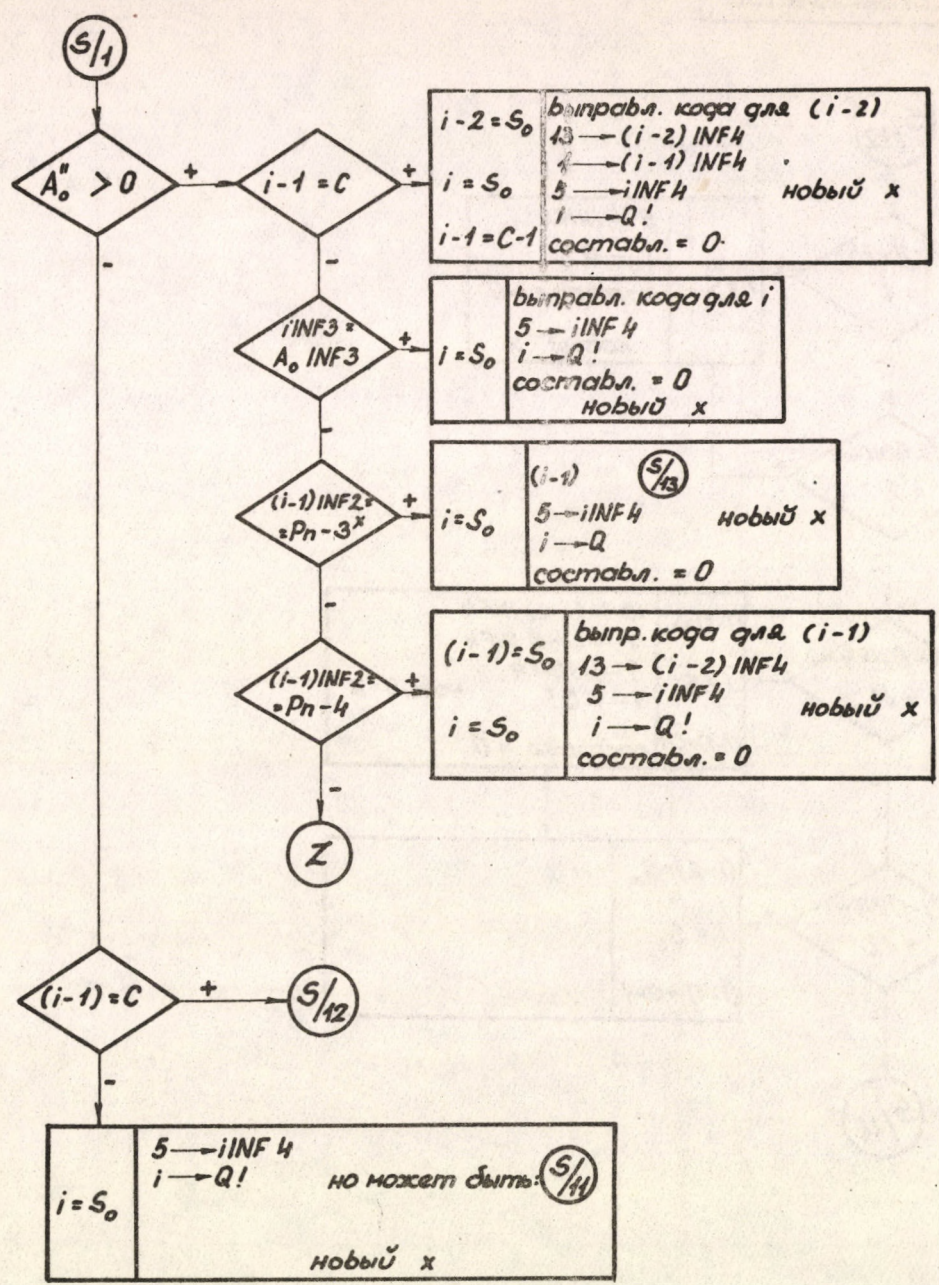


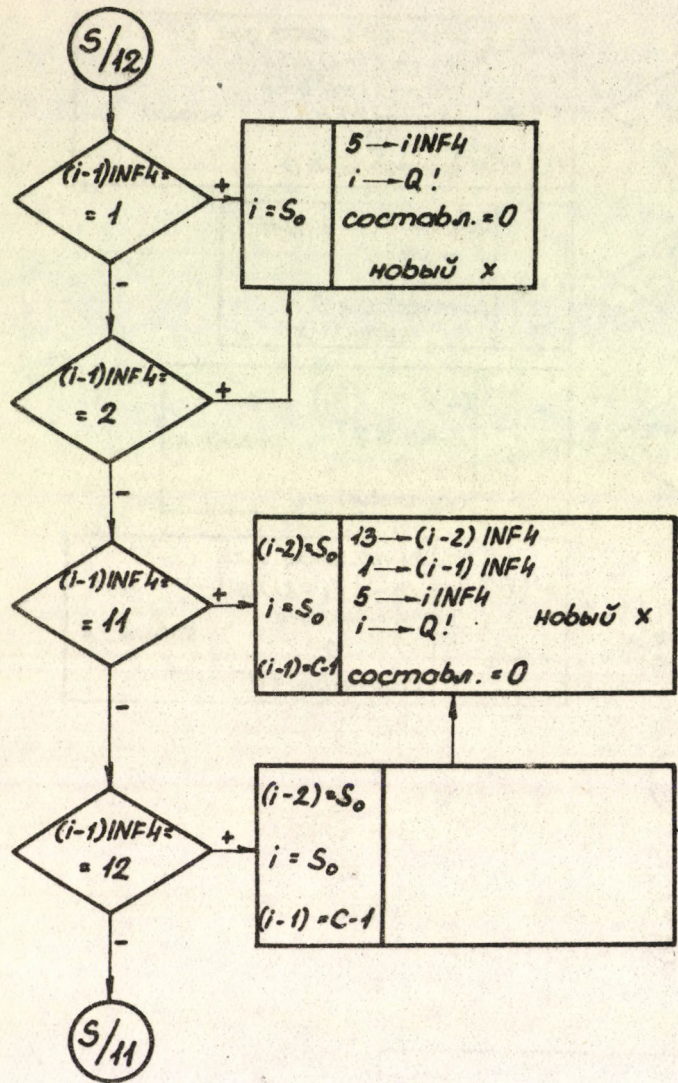




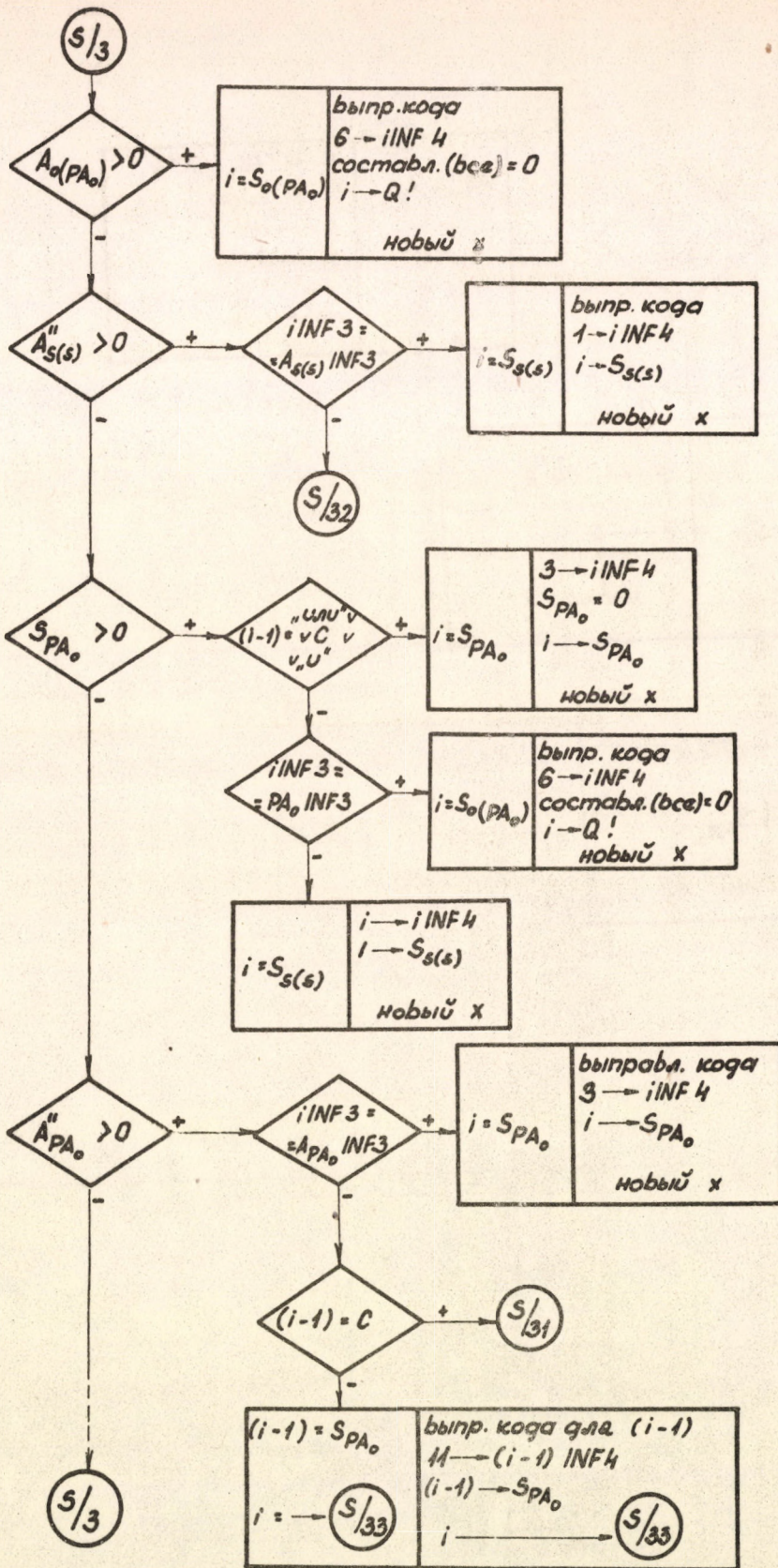
По таблицам В IV.  
Q! Z=0, Y→Z, X→Y, Q→X, i→Q



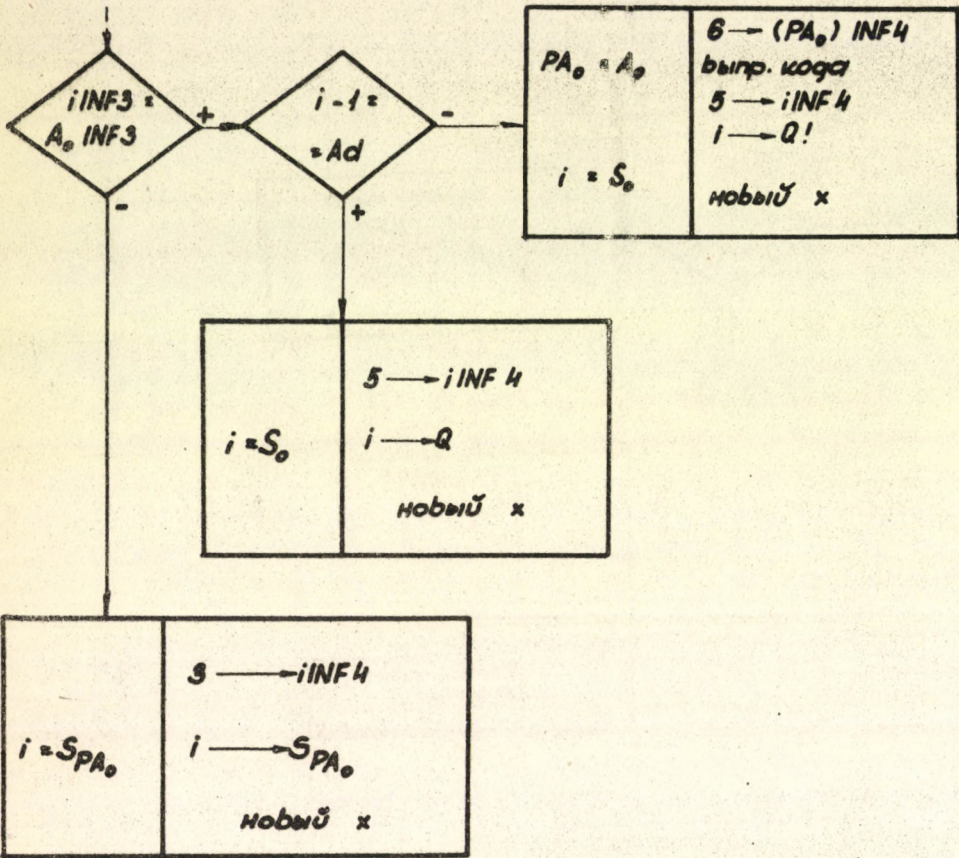






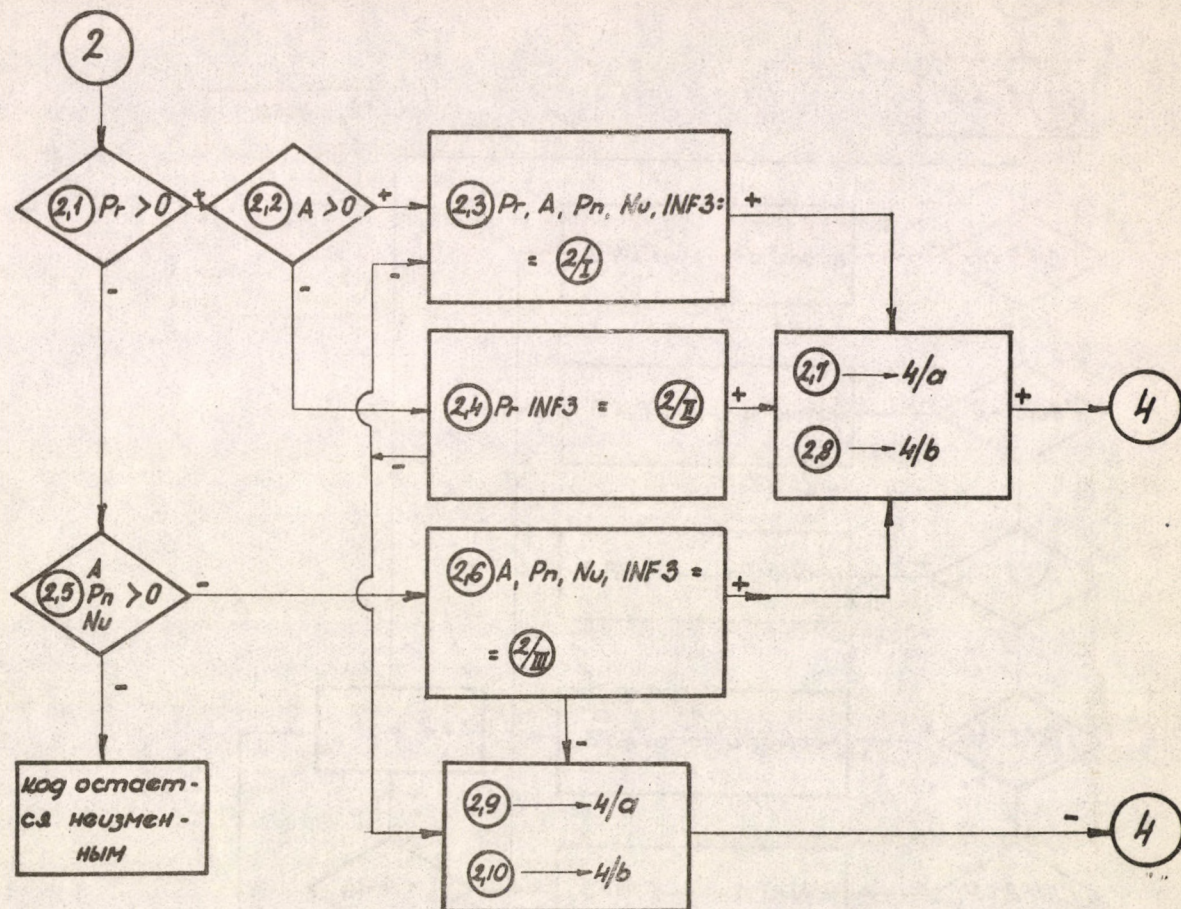


S/3



ПРОГРАММЫ ДЛЯ ВЫПРАВЛЕНИЯ КОДА В НОМИНАЛЬНОЙ ГРУППЕ

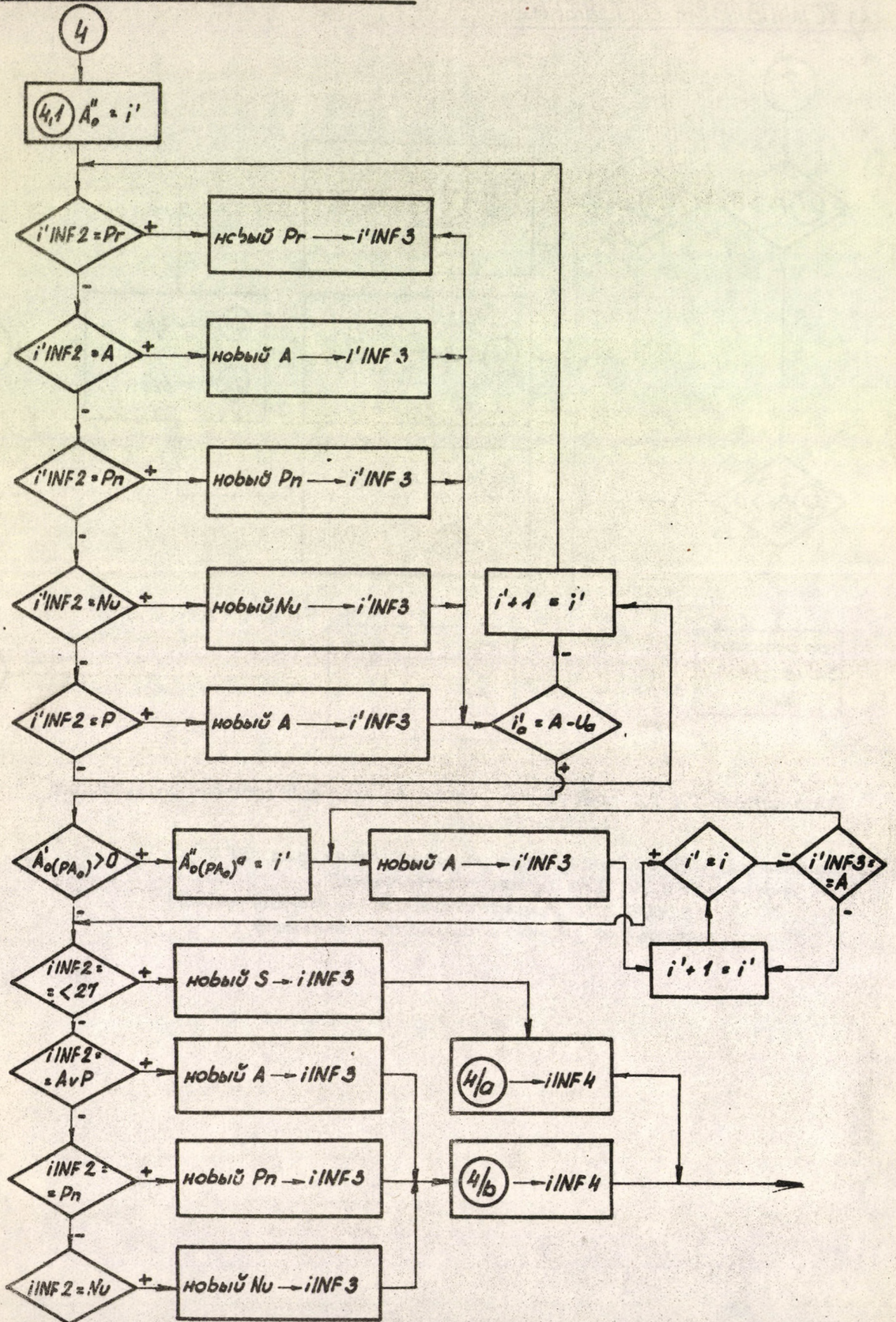
А) УСТАНОВЛЕНИЕ СОГЛАШЕНИЯ



Параметры 2,1-2,6 получают значения индексов соответствующих составляющих

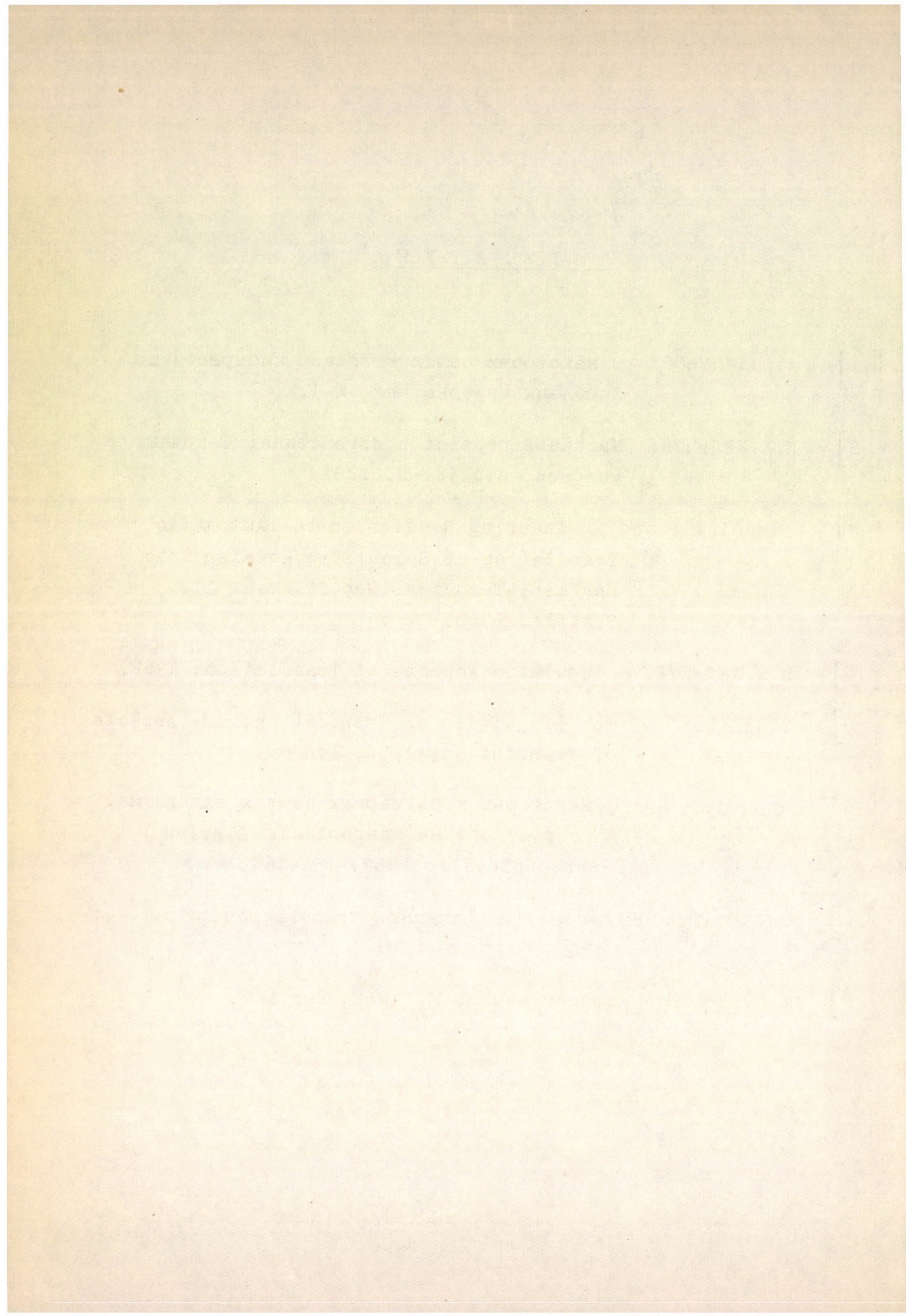
Параметры 2,7-2,10 получают кодовые числа для INF4  
 причем 2,7 и 2,9 для существительных,  
 2,8 и 2,10 для прилагательных и причастий,  
 имеющих значение существительного

Б) ВЫПРАВЛЕНИЕ КОДОВЫХ ЧИСЕЛ



ЛИТЕРАТУРА

- [ 1 ] А.А.ЛЯПУНОВ: О некоторых общих вопросах кибернетики; Проблемы кибернетики, № 1.
- [ 2 ] И.А.МЕЛЬЧУК: Машинный перевод и лингвистика; О точных методах, стр.42. М. 1961.
- [ 3 ] Linguistic and Engineering Studies in the Automatic Translation of Scientific Russian into English; Technical Report Phase II. Seattle, 1960.
- [ 4 ] N.D.ANDREYEV: Linguistic Aspects of Translation; 1962.
- [ 5 ] C.RABIN: The Linguistic of Translation. In: Aspects of Translation; 1958, London
- [ 6 ] HELL,Gy.: О некоторых характерных чертах алгоритма МИ с русского на венгерский. Slavica Debreceniensis, 1962. 151-164.
- [ 7 ] A.G.OETTINGER: Automatic Language Translation, стр.144.; 1960.
- [ 8 ] И.И.РЕВЗИН: Модели языка; М. 1962; стр.155.



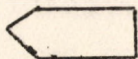
# ALGORITHMS FOR ANALYSIS OF HOMONYM NOMINAL FORMS

By Győző Sipőczy

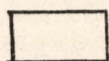
Symbols and abbreviations:



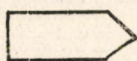
= N = Noun



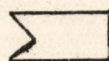
= A = Adjective



= Nu = Numeral



= P = Pronoun



= Pr = Preposition

m = masculine  
f = feminine  
n = neuter  
s = singular  
p = plural

NC = Nominative Case  
AC = Accusative Case  
GC = Genitive Case  
DC = Dative Case  
IC = Instrumental Case  
PC = Prepositional Case

## 1. Coding of the Information

In Russian-Hungarian machine translation - according to the program in preparation - the morphological and syntactical analysis of the word forms of the source language is carried out on the basis of a special code system<sup>1/</sup> set in the addresses of the machine dictionary. This code system contains three code numbers consisting of 6 bits each: the first one gives information on the form class of the morpheme given in the dictionary, the second contains the morphological data, the third unit consisting of 6 bits informs on the semantic distinctives necessary to the analysis.<sup>2/</sup> The form class and the semantic distinctives being unchanging characteristics of the morpheme, the first and third code numbers can be put in the memory together with the addresses of the dictionary.

The second code number, indicating morphological data, depends on the word form given in the text to be translated and so it can be settled only after analysing the given form. This work is to be done by the computer itself with the help of morphological informations put in the grammatical storage and algorithms necessary to make it able to perform various operations (cutting endings, comparing, etc.).

In the following instead of describing which case - or cases - the nominal form can denote, for the sake of shortness code numbers are used for substantives in the figures given below and as indices of the abbreviations. For analysing the homonym nominal forms the following code numbers are necessary:

---

1/ The code system has been worked out by George Hell.

2/ The semantic code system has not been worked out yet.



Noun:

- |                       |                                  |                          |
|-----------------------|----------------------------------|--------------------------|
| 1/ m, s, NC;          | 32/ p, NC, AC;                   | 18/ n, s, PC;            |
| 2/ m, s, AC;          | 33/ p, AC, GC;                   | 19/ p, NC;               |
| 3/ m, s, GC;          | 34/ n, s, GC; p, NC, AC;         | 20/ p, AC;               |
| 4/ m, s, DC;          | 35/ m, s, PC; p, NC, AC;         | 21/ p, GC;               |
| 5/ m, s, IC;          | 36/ f, s, GC; p, NC, AC;         | 22/ p, DC;               |
| 6/ m, s, PC;          | 37/ f, s, GC, DC, PC; p, NC, AC; | 23/ p, IC;               |
| 7/ f, s, NC;          | 38/ m, s, AC, GC; p, NC;         | 24/ p, PC;               |
| 8/ f, s, AC;          | 39/ m, s, GC; p, NC, AC;         |                          |
| 9/ f, s, GC;          | 40/ f, s, NC, AC;                |                          |
| 10/ f, s, DC;         | 41/ f, s, GC; p, NC;             | 25/ m, s, NC, AC;        |
| 11/ f, s, IC;         | 42/ n, s, AC, PC;                | 26/ m, n, s, AC, GC;     |
| 12/ f, s, PC;         | 43/ f, s, PC; p, AC;             | 27/ m, n, s, DC;         |
| 13/ n, s, NC;         |                                  | 28/ m, n, s, PC;         |
| 14/ n, s, AC;         | <u>Adjective (Numeral,</u>       | 29/ f, s, GC DC, IC, PC; |
| 15/ n, s, GC;         | <u>Pronoun):</u>                 | 30/ n, s, NC, AC;        |
| 16/ n, s, DC;         | 1/ m, s, NC;                     | 31/ p, AC, GC, PC;       |
| 17/ n, s, IC;         | 2/ m, s, AC;                     | 32/ p, NC, AC;           |
| 18/ n, s, PC;         | 3/ m, s, GC;                     | 33/ m, n, s, IC; p, DC;  |
| 19/ p, NC;            | 4/ m, s, DC;                     | 34/ Comparative Deg-     |
| 20/ p, AC;            | 5/ m, s, IC;                     | ree; simple              |
| 21/ p, GC;            | 6/ m, s, PC;                     | form <sup>1/</sup>       |
| 22/ p, DC;            | 7/ f, s, NC;                     | 35/ m, s, NC, AC; f, s,  |
| 23/ p, IC;            | 8/ f, s, AC;                     | GC, DC, IC, PC;          |
| 24/ p, PC;            | 9/ f, s, GC;                     | 36/ f, AC, GC;           |
|                       | 10/ f, s, DC;                    | 37/ f, s, DC, PC;        |
|                       | 11/ f, s, IC;                    | 38/ m, n, s, IC;         |
|                       | 12/ f, s, PC;                    | 39/ f, s, GC, IC;        |
|                       | 13/ n, s, NC;                    | 40-44/ Predicati-        |
|                       | 14/ n, s, AC;                    | ve forms <sup>2/</sup>   |
|                       | 15/ n, s, GC;                    | 45/ p, AC, GC;           |
|                       | 16/ n, s, DC;                    | 46/ f, s, GC, DC;        |
|                       | 17/ n, s, IC;                    |                          |
|                       |                                  |                          |
|                       |                                  |                          |
| 25/ m, s, NC, AC;     |                                  |                          |
| 26/ m, s, DC, PC;     |                                  |                          |
| 27/ m, s, AC, GC;     |                                  |                          |
| 28/ f, s, DC, PC;     |                                  |                          |
| 29/ f, s, GC, DC;     |                                  |                          |
| 30/ n, s, NC, AC;     |                                  |                          |
| 31/ n, s, NC, AC, PC; |                                  |                          |

1/ It has no role in the analysis of the homonym nominal forms.

2/ They have no role in the analysis of the homonym nominal forms.

Cardinal Number:

34/ m,n,s,NC AC;

35/ NC,AC;

36/ AC,GC,PC;

37/ DC;

38/ IC;

39/ m,NC;

40/ n,NC;

41/ m,AC;

42/ n,AC;

43/ f,NC,AC;

44/ f,NC;

45/ f,AC;

46/ GC;

47/ PC;

48/ GC,DC,IC,PC;

49/ NC;

50/ AC;

51/ AC,GC;

Preposition:

1/ governing only GC;

2/ governing only DC;

3/ governing only AC;

4/ governing only IC;

5/ governing only PC;

---

---

6/ governing AC and IC;

7/ governing AC and PC;

8/ governing DC and PC;

9/ governing GC and IC;<sup>1/</sup>

The code numbers below the double line in the groups (in the group "Cardinal Number" the ones with the frame) denote homonym forms. The description necessary for them is given in point 2 (p.5-8). In the following parts of the present work the letter

---

<sup>1/</sup> The preposition "C" belonging here governs accusative case, too, but in technical texts this case practically never occurs, so the genitive and instrumental cases only are taken into account here.

- denoting the kind of the substantive and the code number - index of the given abbreviation, or the symbol and the code number in it, are used together as signs of the substantive forms. E.g.: N<sub>28</sub> =  $\diamond 28$  = noun form of the type "машине";

A<sub>33</sub> =  $\square 33$  = adjective form of the type "новым, хорошим".

## 2. Homonym Forms

One of the difficulties of Russian Hungarian Machine Translation arises from the fact that the endings of the substantives are not always only for one case, and so the second code number determined by the computer does not give a base necessary to find out the Hungarian equivalent of the given word form. Therefore, in the code system mentioned above the homonym endings are indicated by separate code numbers /beginning from 25/ and if one of these gets into the computer, the role of the ending in the given sentence is to be found by further analysis. This analysis necessitates special programs, by means of which the computer finds the role given in the sentence and changes the original code number - denoting several cases - for another one - denoting only one case. This changed code number gives the sufficient information to define the real sense of the given word form and to find its equivalent in the target language.

Code changing programs are necessary in the following cases:

### I. Nouns:

N<sub>25</sub>: the forms of dictionary of nouns of masculine gender, denoting inanimate things, e.g.: завод, трамвай, камень, etc. The form is for singular nominative and accusative /masculine/.

N<sub>26</sub>: the form with the ending "-у" of nouns of the "лес, мост" type, e.g.: лесу, мосту, etc. The form is for singular dative and prepositional /masculine/.

- N<sub>27</sub>: the form with the ending "-a" of nouns of masculine gender, denoting living beings, e.g.: работника, etc. The form is for singular accusative and genitive cases (masculine).
- N<sub>28</sub>: the form with the ending "-e" of nouns of the "машина" type, e.g.: машина etc. The form is for singular dative and prepositional (feminine).
- N<sub>29</sub>: the code number denotes feminine singular genitive and dative. There is no actual form corresponding to these cases, but at the end of some code changing programs - as it will be seen below - we get such a result. In this case further analysis is necessary.
- N<sub>30</sub>: the form of dictionary of nouns of the "место" type, e.g.: место, производство, etc. The form is for singular nominative and accusative (neuter).
- N<sub>31</sub>: the form of dictionary of nouns of the "поле" type, e.g.: поле, море, etc. The form is for singular nominative, accusative and prepositional (neuter).
- N<sub>32</sub>: the forms with the endings -ы and -и of nouns of masculine gender and that with the ending -а of the nouns of the "время" type, e.g.: паровозы, трамваи, времена, etc. The forms are for plural nominative and accusative (masculine and neuter).
- N<sub>33</sub>: the forms with the ending -ов of the nouns of masculine gender denoting living beings and that with the ending @ of the nouns of feminine gender denoting living beings, e.g.: инженеров, коров, etc. The forms are for plural accusative and genitive (masculine and feminine).
- N<sub>34</sub>: the form with the ending -я of nouns of the "поле" type, e.g.: поля etc. The form is for singular genitive; plural nominative and accusative.. (neuter).
- N<sub>35</sub>: the form with the ending -ии of nouns ending with -ий e.g.: алюминии, калии, etc. The form is for singular prepositional; plural nominative and accusative (masculine).

- N<sub>36</sub>: the form with the ending -и of nouns of the "книга" type, e.g.: книга, etc. The form is for singular genitive, plural nominative and accusative (feminine).
- N<sub>37</sub>: the form with the ending -и of nouns of the "ночь" type, e.g.: ночи, etc. The form is for singular genitive, dative and prepositional; plural nominative and accusative (feminine).
- N<sub>38</sub>: the form with the ending -а of nouns of the "доктор" type denoting living beings, e.g.: доктора, etc. The form is for singular accusative and genitive; plural nominative (masculine).
- N<sub>39</sub>: the form with the ending -а of nouns of the "город" type denoting inanimate things, e.g.: города, etc. The form is for singular genitive; plural nominative and accusative (masculine).
- N<sub>40</sub>: the form of dictionary of the nouns of the "ночь" type; e.g.: ночь, дверь, etc. The form is for singular nominative and accusative (feminine).
- N<sub>41</sub>: the form with the ending -ы of nouns of the "сестра" type denoting living beings and that with the ending -и of nouns of the "девушка" type denoting living beings, e.g.: сестры, девушки, etc. The form is for singular genitive and plural nominative (feminine).
- N<sub>42</sub>: the code number denotes neuter singular accusative and prepositional. Result of some code changing programs.
- N<sub>43</sub>: the code number denotes feminine singular prepositional and plural accusative. Result of some code changing programs.

## II. Adjectives:

- A<sub>25</sub>: the form with the endings -ий, -ий of the adjectives, e.g.: новый, хороший, etc. The forms are for singular nominative and accusative (masculine).
- N<sub>26</sub>: the form with the endings -ого, его of the adjectives, e.g.: нового, хорошего, etc. The forms are for singular accusative and genitive (masculine and neuter).
- N<sub>29</sub>: the form with the ending -ой of adjectives of the type "новый" and those with the endings -ой, -ей of adjectives of the "хороший" and "летний" type. The forms are for singular genitive, dative, instrumental and prepositional (feminine).
- A<sub>30</sub>: the forms with the endings -ое, -ее of the adjectives, e.g.: новое, летнее, etc. The forms are for singular nominative and accusative (neuter).
- A<sub>31</sub>: the forms with the endings -ых, их of the adjectives, e.g.: новых, хороших, etc. The forms are for plural accusative, genitive and prepositional.
- A<sub>32</sub>: the forms with the endings -ые, -ие of the adjectives, e.g.: новые, хорошие, etc. The forms are for plural nominative and accusative.
- A<sub>33</sub>: the forms with the endings -ым, -им of the adjectives, e.g.: новым, хорошим, etc. The forms are for singular instrumental (masculine and neuter) and plural dative.
- A<sub>35</sub>: the form with the ending -ой of adjectives of the "большой" type, e.g.: большой etc. The forms are for singular nominative and accusative (masculine); singular genitive, dative, instrumental and prepositional (feminine).
- A<sub>37</sub>: the code number denotes singular dative and prepositional (feminine). Result of some code changing programs.

As the morphological features of the pronouns and numerals having a part in nominal word groups are the same as those of the adjectives, the indices of the adjectives are to be found with the abbreviation P and partly with the abbreviation Nu as well.

### III. Numerals:

For the numeral "один" see the adjectives.

Nu<sub>34</sub>: два: singular nominative and accusative (masculine neuter).

Nu<sub>35</sub>: три, четыре, пять, сто, сорок and the other numerals of the same type: nominative and accusative.

Nu<sub>36</sub>: двух, трёх, четырёх: accusative, genitive and prepositional.

Nu<sub>43</sub>: две: nominative and accusative (feminine).

Nu<sub>48</sub>: пяти, сорока and the other corresponding forms of numerals of the same type: genitive, dative, instrumental and prepositional.

Nu<sub>51</sub>: двухсот and the other corresponding forms of numerals of the same type: accusative and genitive.

### 3. Types of the Nominal Word Groups

When analysing the nominal forms listed above it is necessary to examine the other substantives forming a word group with the given nominal form and the preposition in the word group, if any. If the ending of a substantive denotes more than one case, the connection of the given form

with the other members of the word group in the majority of the cases will clearly show the exact role of the form being analysed.

To the analysis of the homonym noun forms the following types of word groups are to be taken into account:

- 1/ Adjective + noun
- 2/ Numeral + noun
- 3/ Pronoun + noun
- 4/ Preposition + noun
- 5/ Numeral + adjective + noun<sup>1/</sup>
- 6/ Pronoun + adjective + noun<sup>1/</sup>
- 7/ Preposition + adjective + noun
- 8/ Numeral + pronoun + noun
- 9/ Pronoun + numeral + noun
- 10/ Preposition + numeral + noun
- 11/ Preposition + pronoun + noun
- 12/ Numeral + pronoun + adjective + noun<sup>1/</sup>
- 13/ Pronoun + numeral + adjective + noun<sup>1/</sup>
- 14/ Preposition + numeral + adjective + noun<sup>1/</sup>
- 15/ Preposition + pronoun + adjective + noun<sup>1/</sup>
- 16/ Preposition + numeral + pronoun + noun
- 17/ Preposition + pronoun + numeral + noun
- 18/ Preposition + numeral + pronoun + adjective + noun<sup>1/</sup>
- 19/ Preposition + pronoun + numeral + adjective + noun<sup>1/</sup>

By listing the word groups that can be formed with each homonym noun form and Prepositions or other substantives, after a proper arrangement, the possibilities offered by the types of word groups enumerated above in clearing the role of the homonym forms in the group will be got. For the sake of shortness we can sum up these possibilities in

---

<sup>1/</sup> The types of word groups in which the adjective precedes the numeral or pronoun are not taken into account partly because such cases are very rare, partly because the order plays no part here in deciding the case of the noun.



figures. So the following figures - on the base of the types of word groups listed above - contain the possible connections of the various types of substantive and prepositional forms. In the symbols of the figures the upper number indicates the original code number of the given substantive or preposition, the lower one denotes the code number showing the result received after considering the relations of the forms in the word group. The number on top of the columns shows the place of the members of the word group. E.g.: 0 = noun; - 1: the member before the noun; - 2: the second member before the noun, etc. The lines connecting the symbols show with which following member the given member can form connection. The connections in which the order of succession of the numeral and pronoun can be changed without the change of the meaning are indicated by  $\longleftrightarrow$ . The full stop inside the symbols is only for separating the single code numbers (e.g.: 3.6 = either 3 or 6).

The lower (changed) code number for numerals, governing genitive case (singular or plural) in the source language will indicate plural nominative and accusative in accordance with the syntactical connections of the given numeral form. This is necessary because of the logical connections of the source language. The same solution is to be found in the algorithm for analysing Russian language worked out by Meltsuk<sup>1/</sup>.

---

<sup>1/</sup> И.А.МЕЛЬЧУК: Два оператора установления соответствия. Москва, 1961. In this work it is to be seen how the "operators" can be used for analysing homonym nominal forms.

Fig. 1.

$N_{25}$

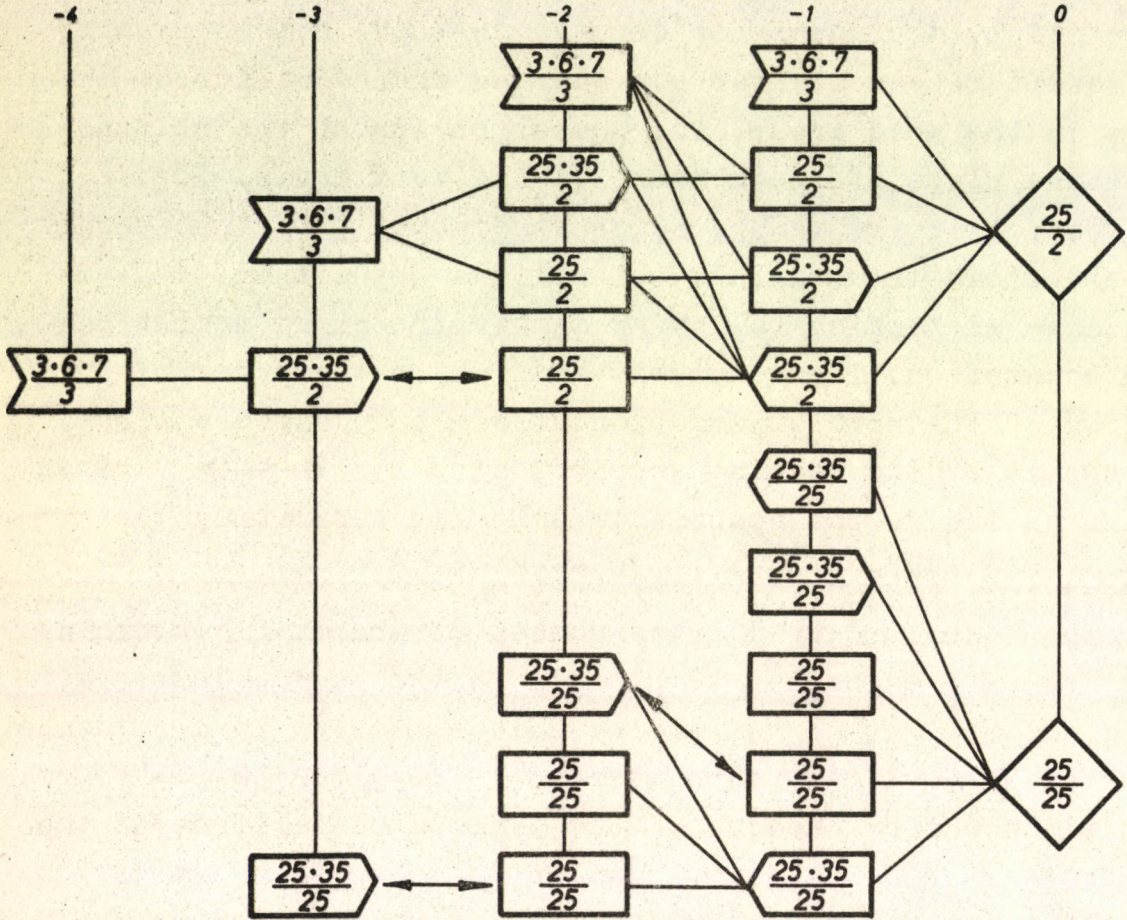


Fig. 2.

$N_{26}$

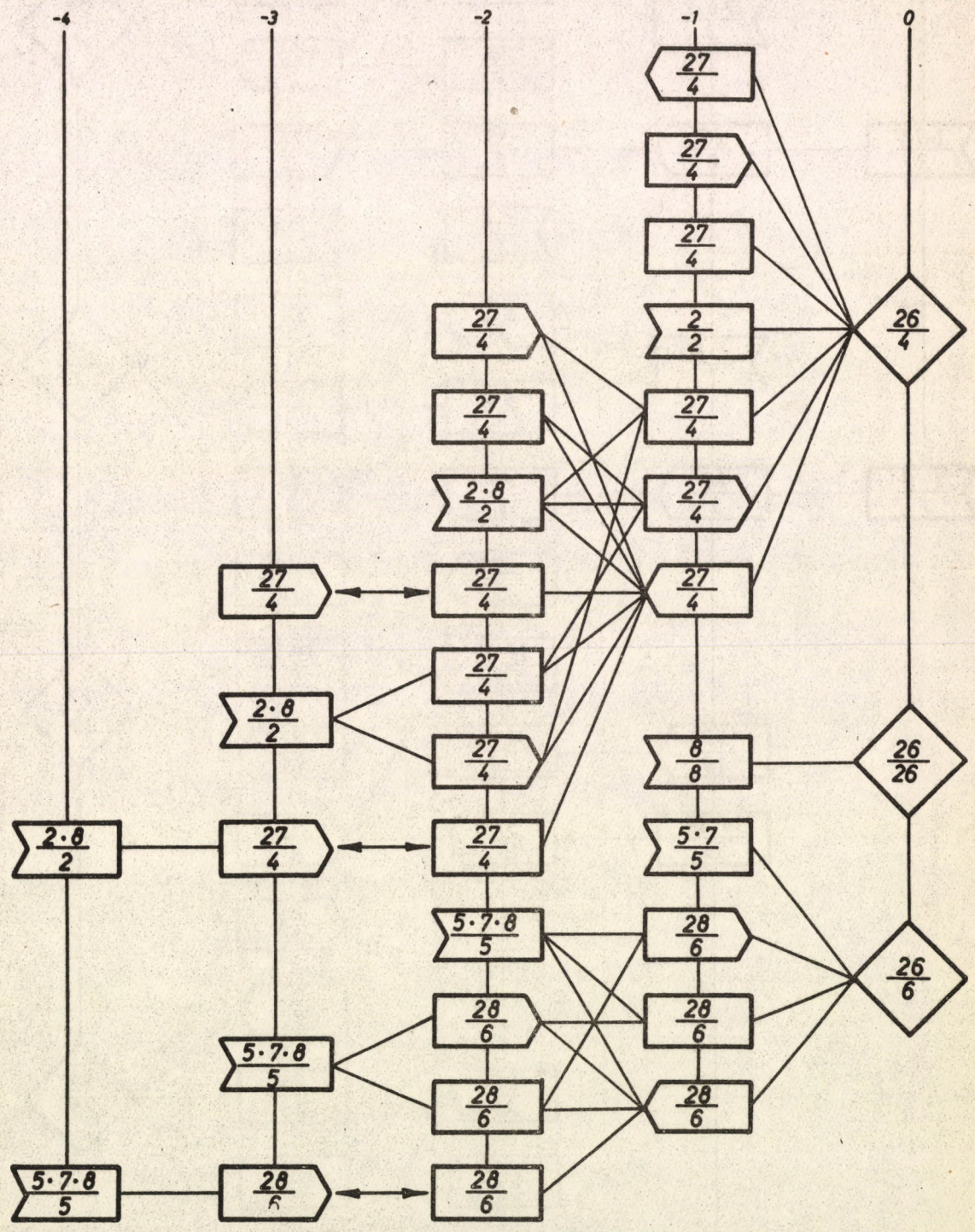


Fig. 3.

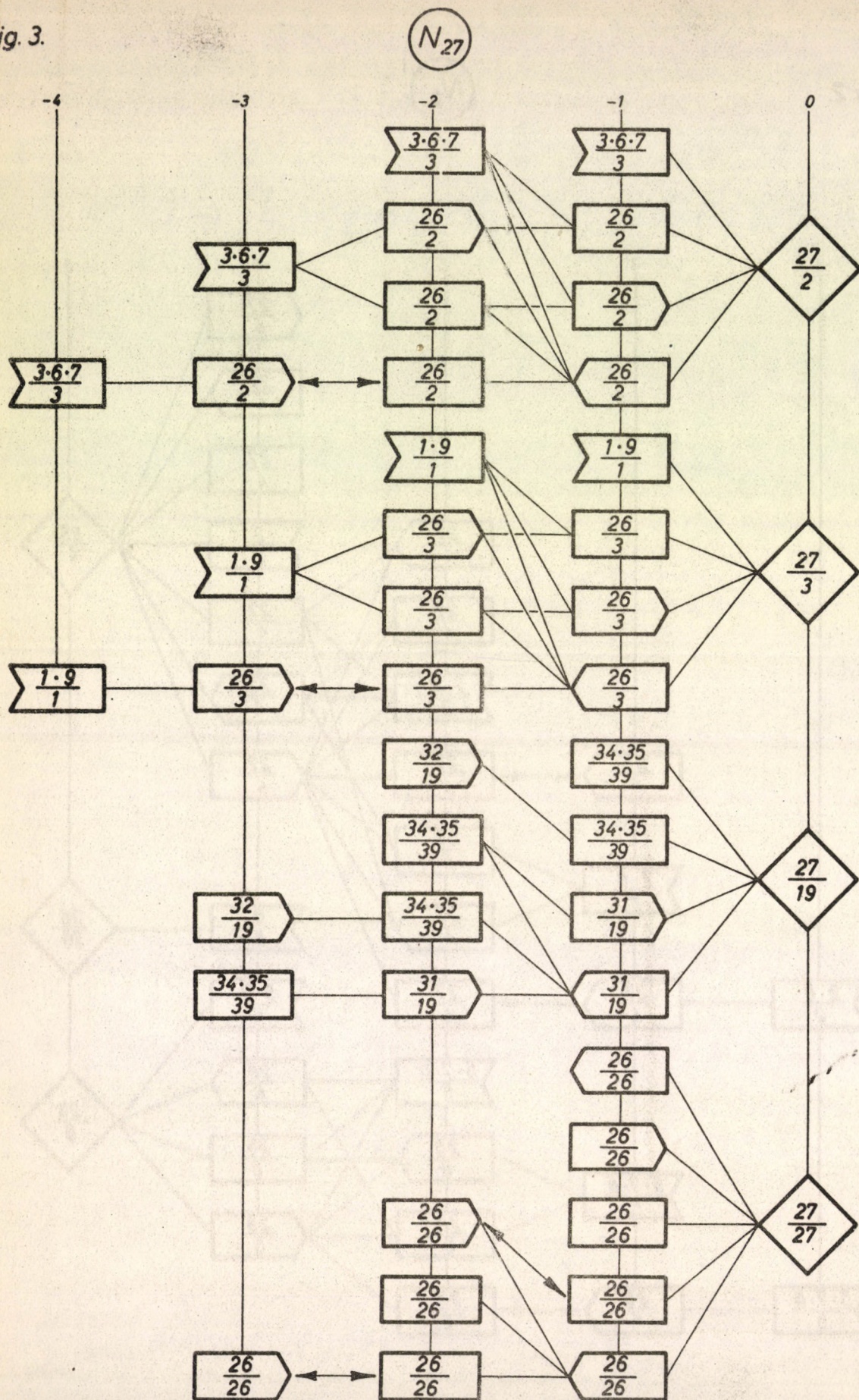


Fig. 4.

$N_{28}$

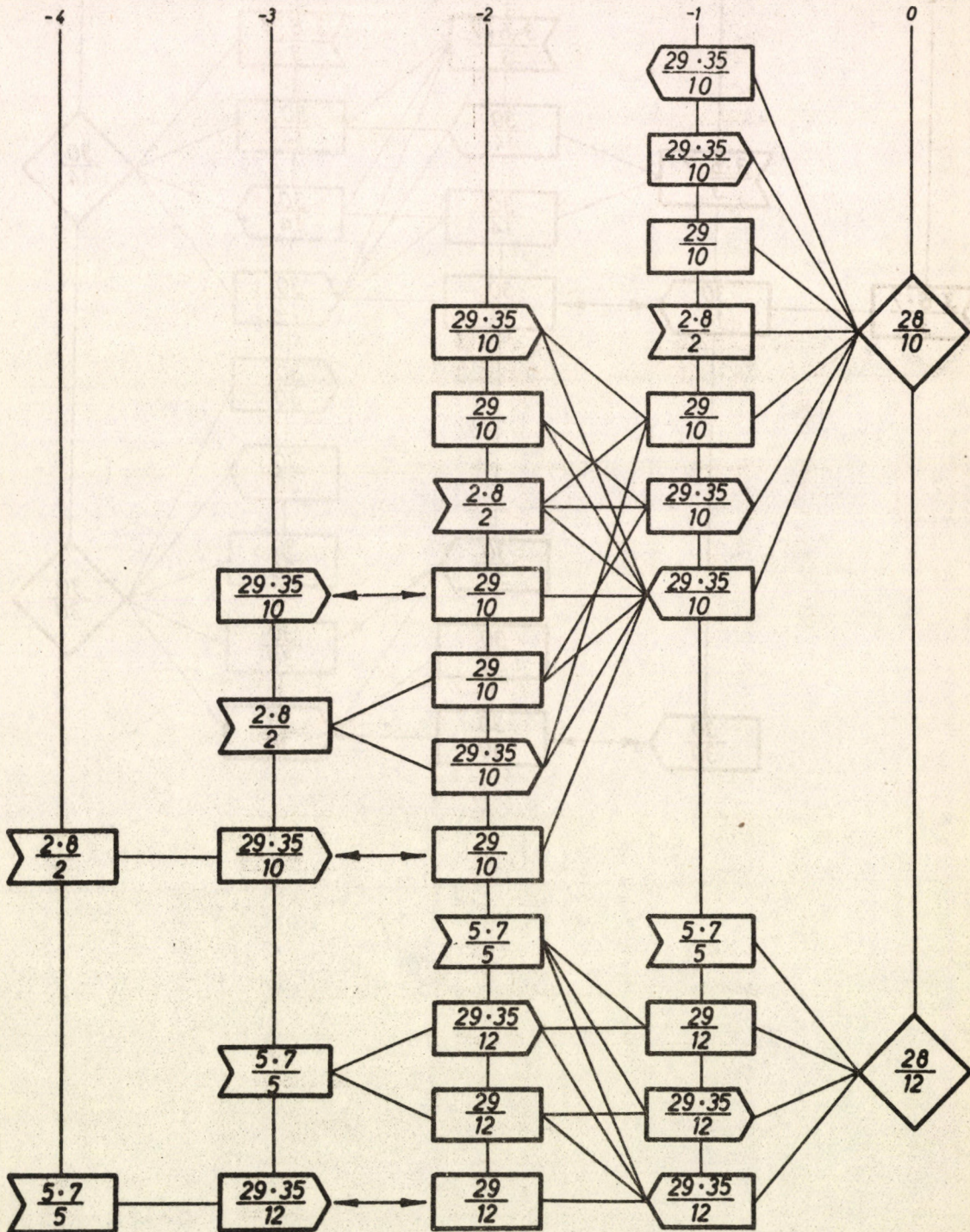


Fig. 5.

$N_{30}$

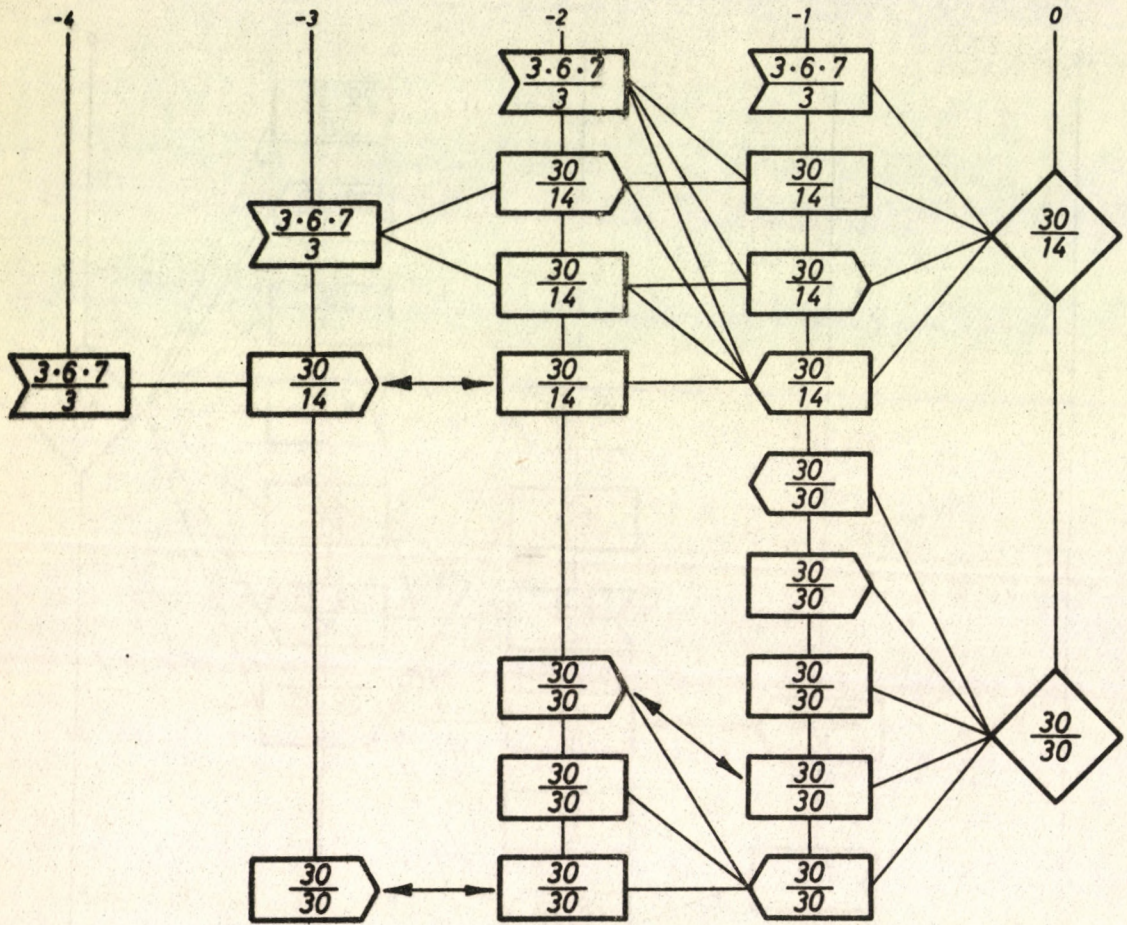


Fig. 6.

$N_{31}$

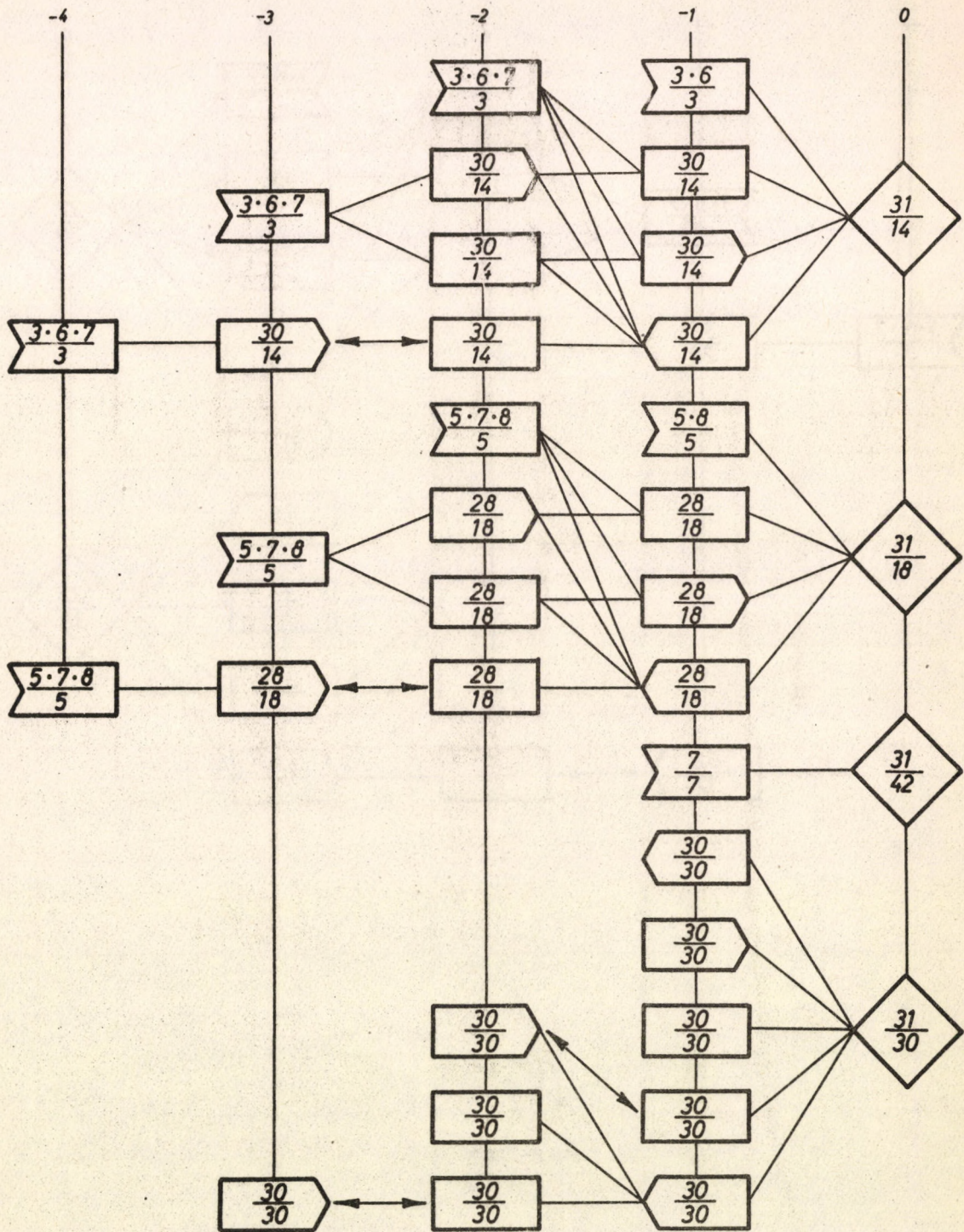


Fig. 7.

$N_{32}$

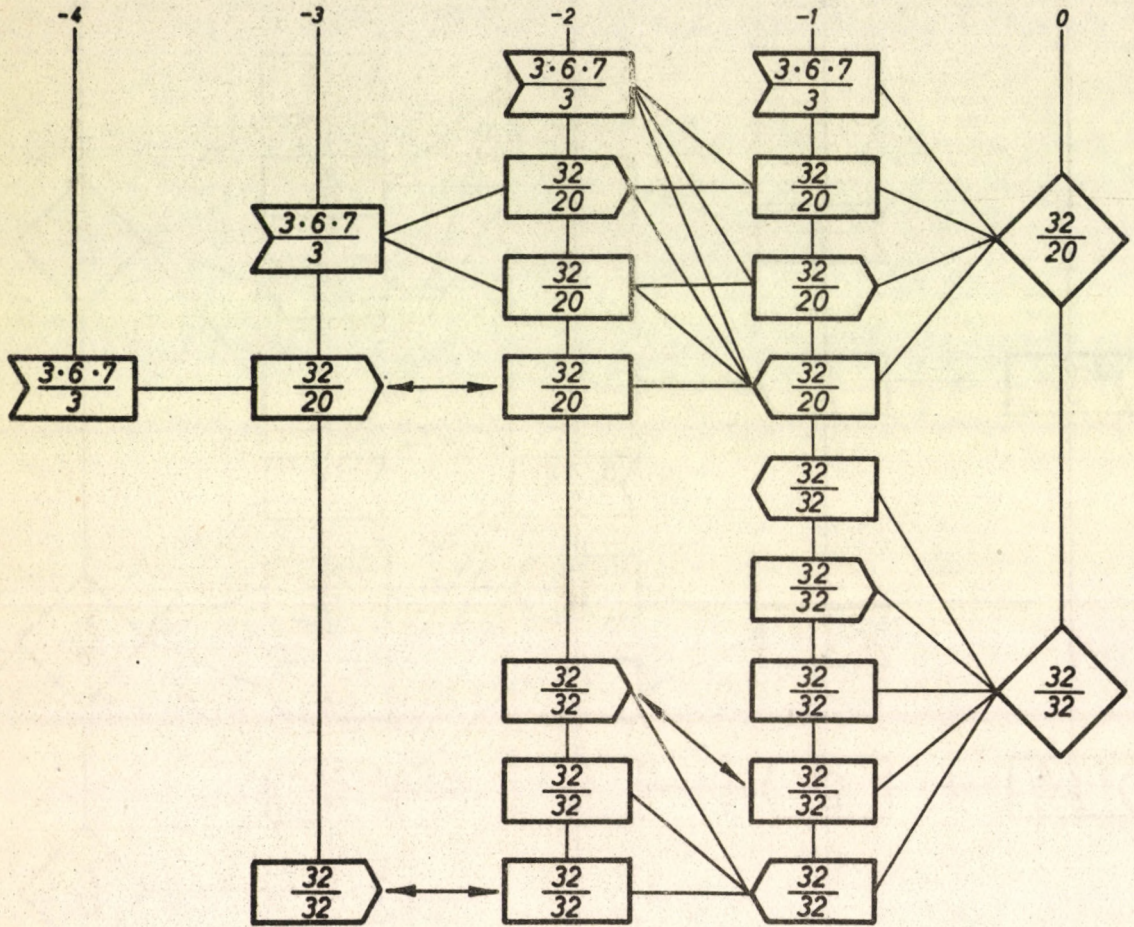




Fig. 8.

$N_{33}$

A

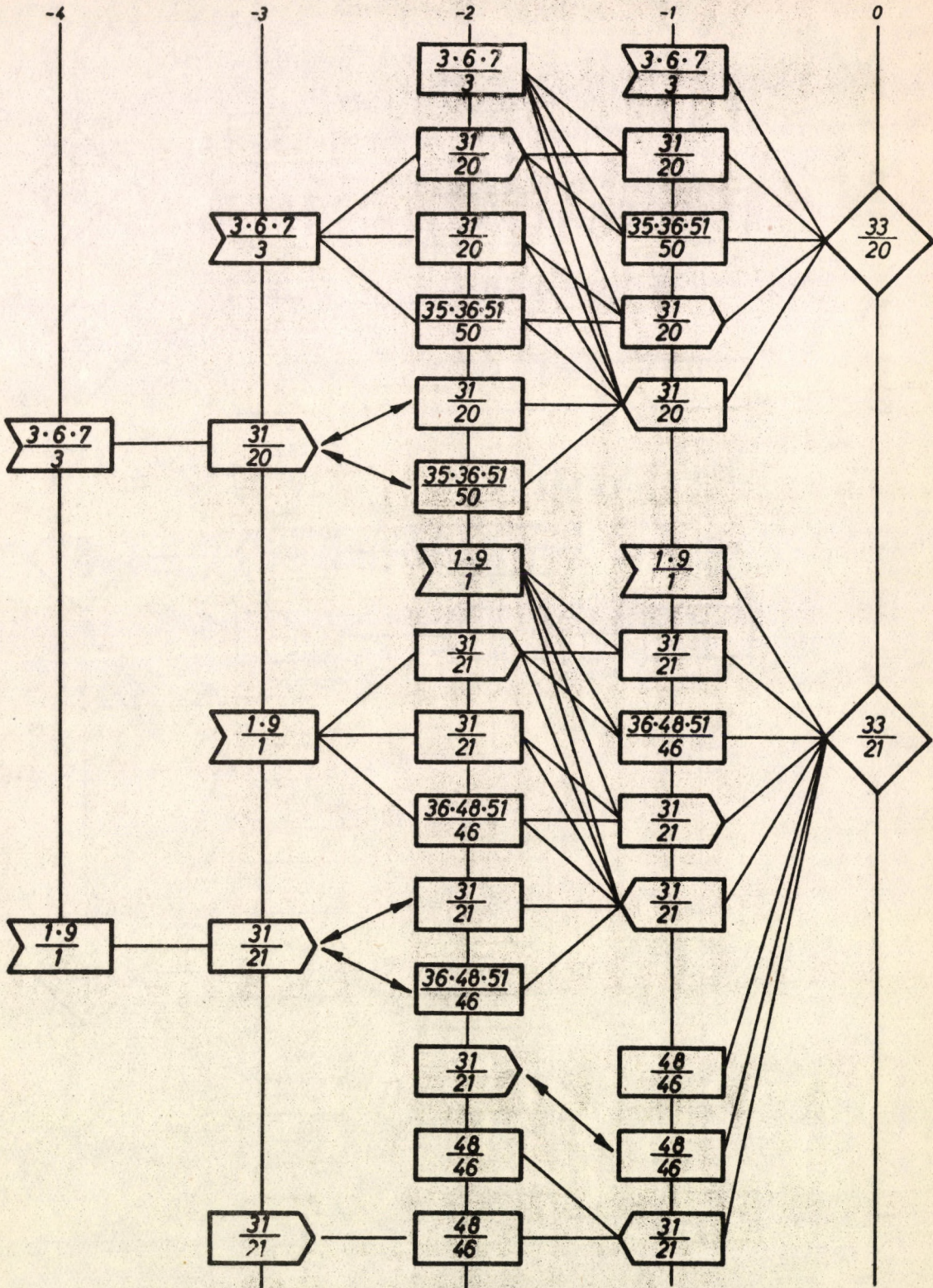


Fig. 8.

$N_{33}$

B

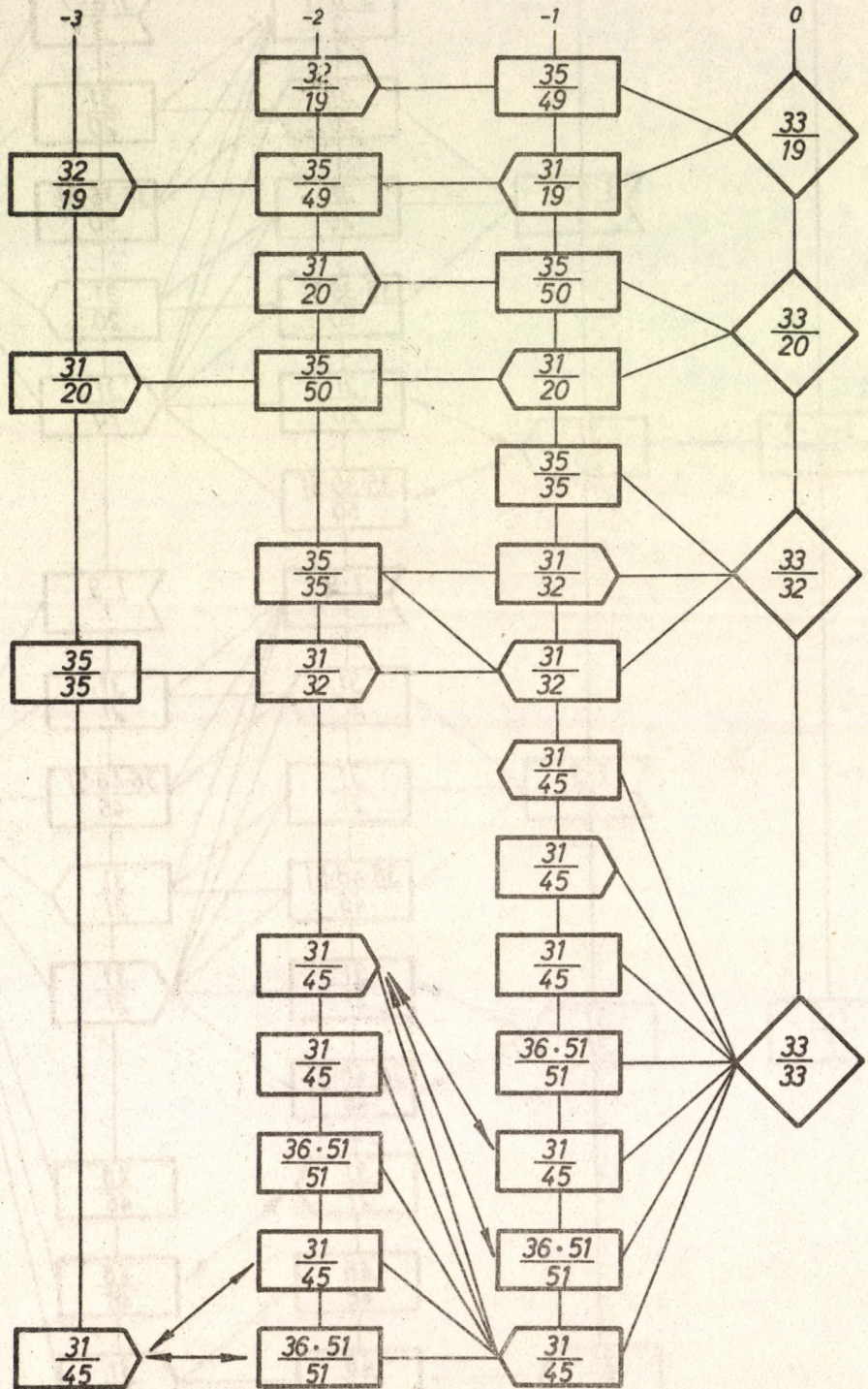


Fig. 9.

$N_{34}$

A

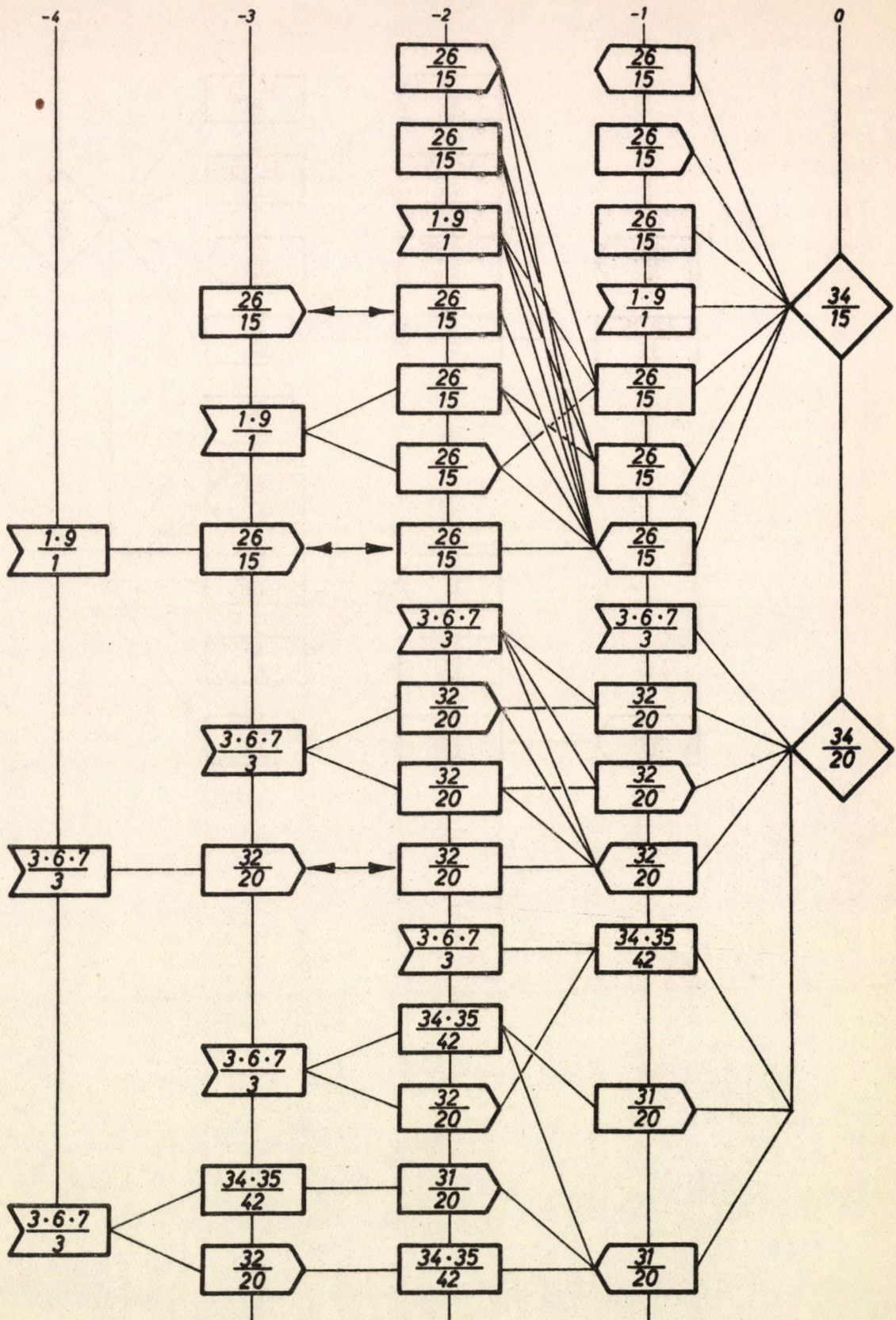


Fig. 9.

$N_{34}$

B

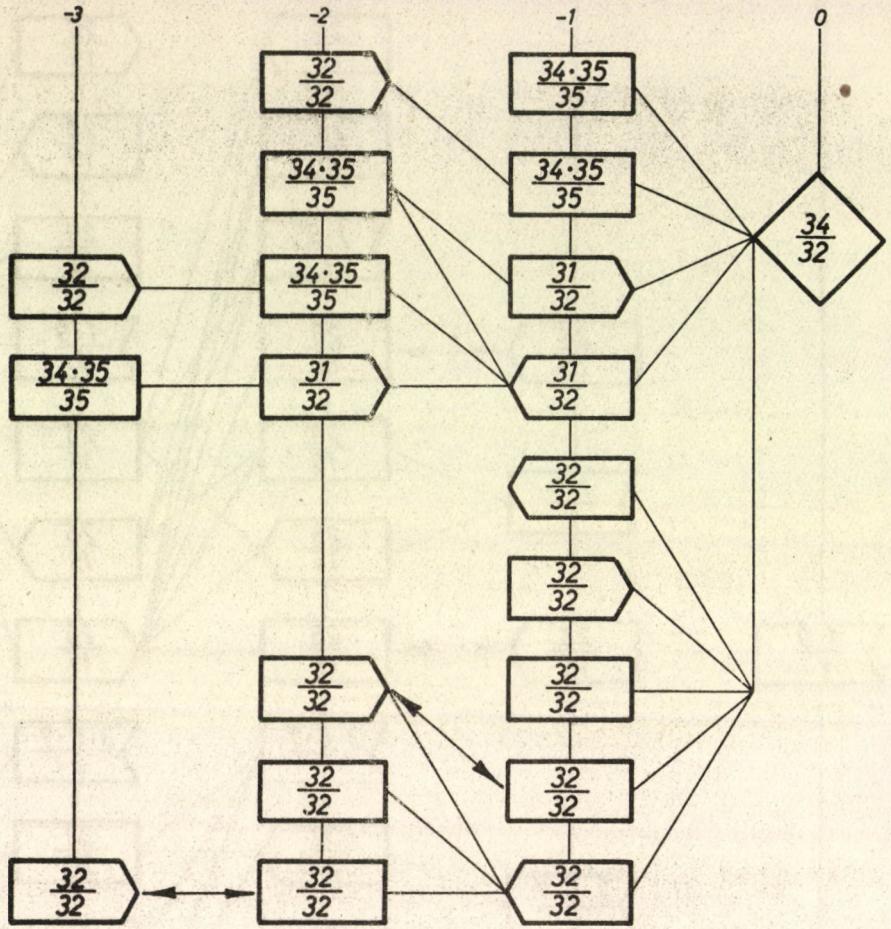


Fig. 10.

$N_{35}$

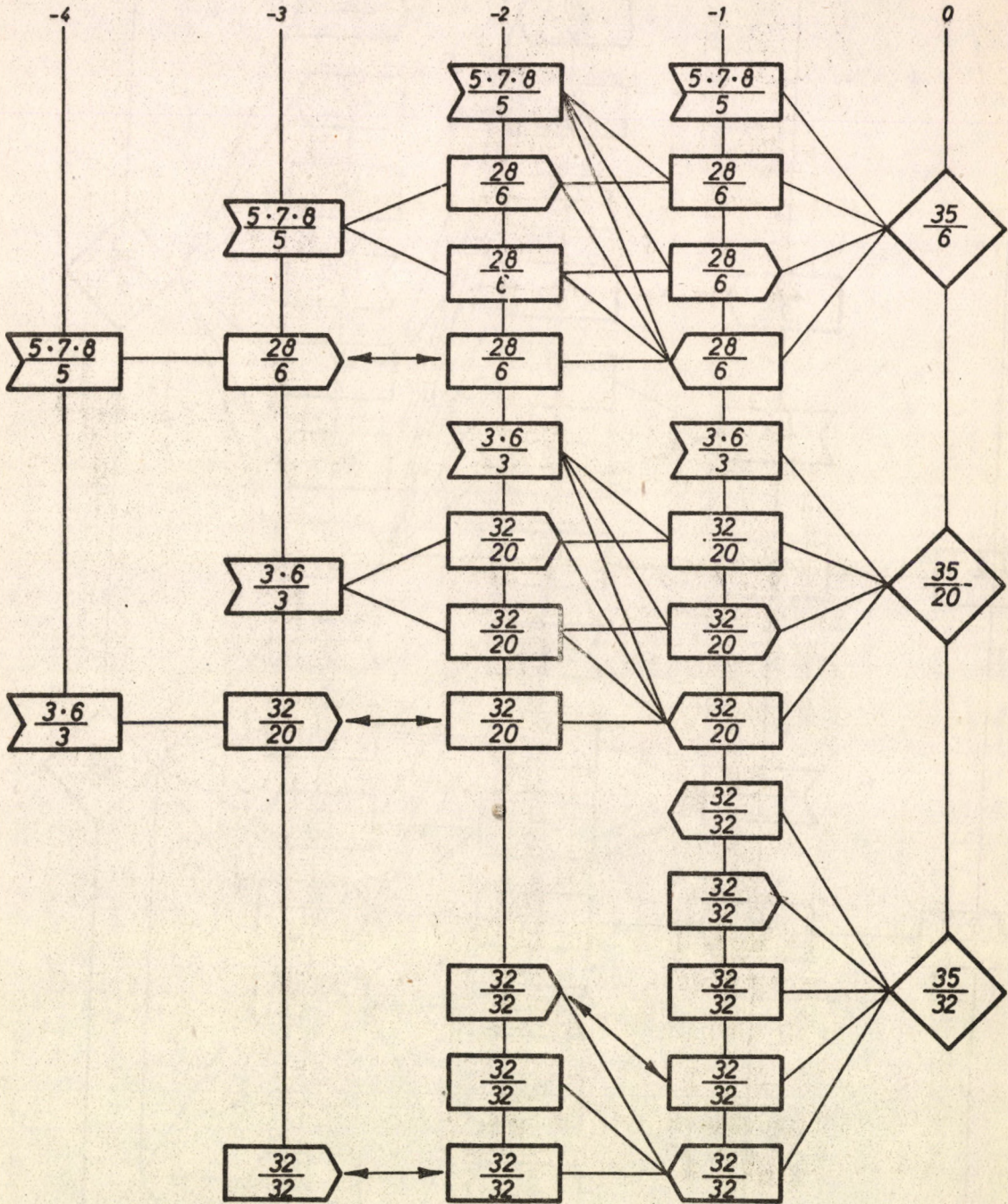


Fig. 11.

$N_{36}$

A

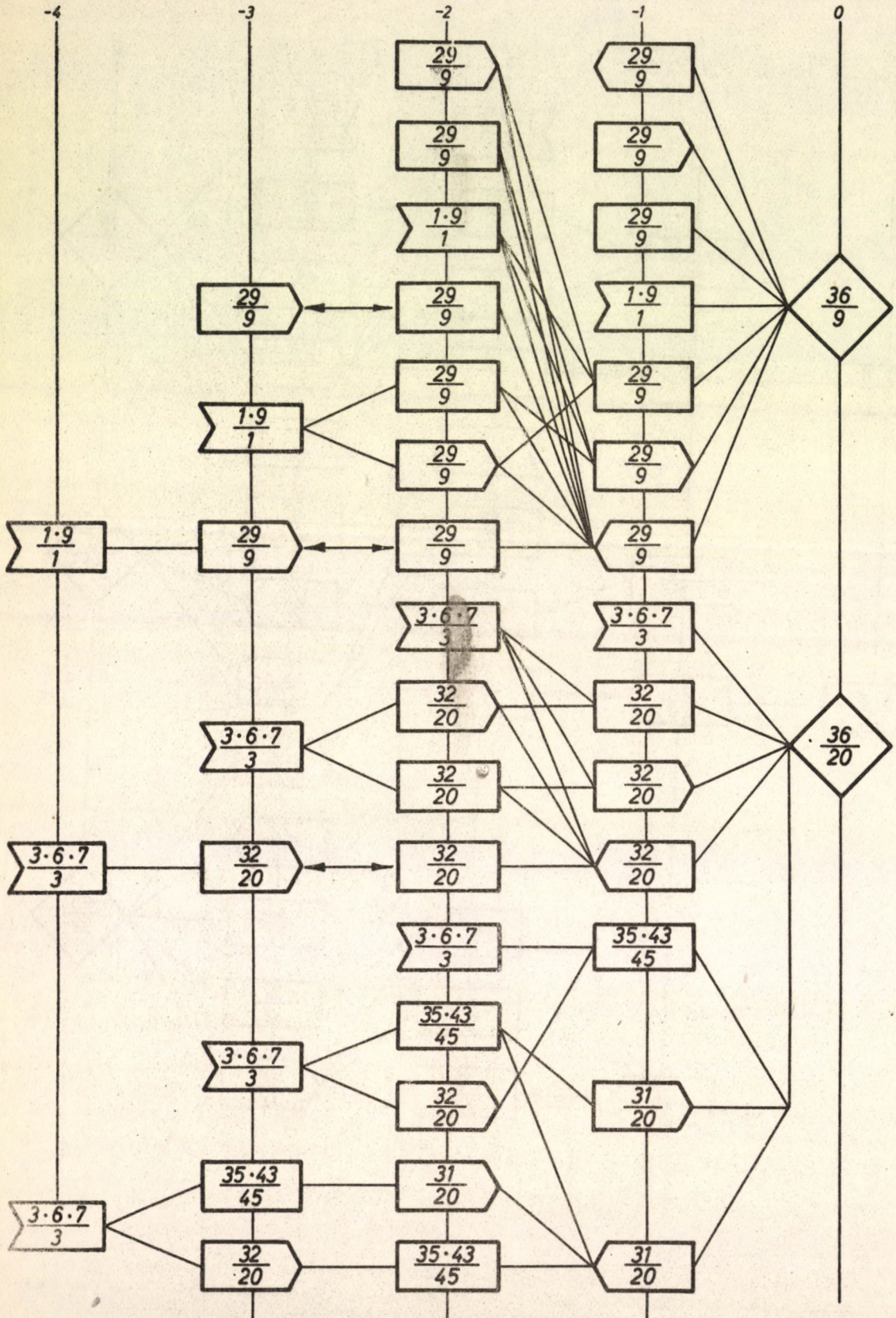


Fig. 11.

$N_{36}$

B

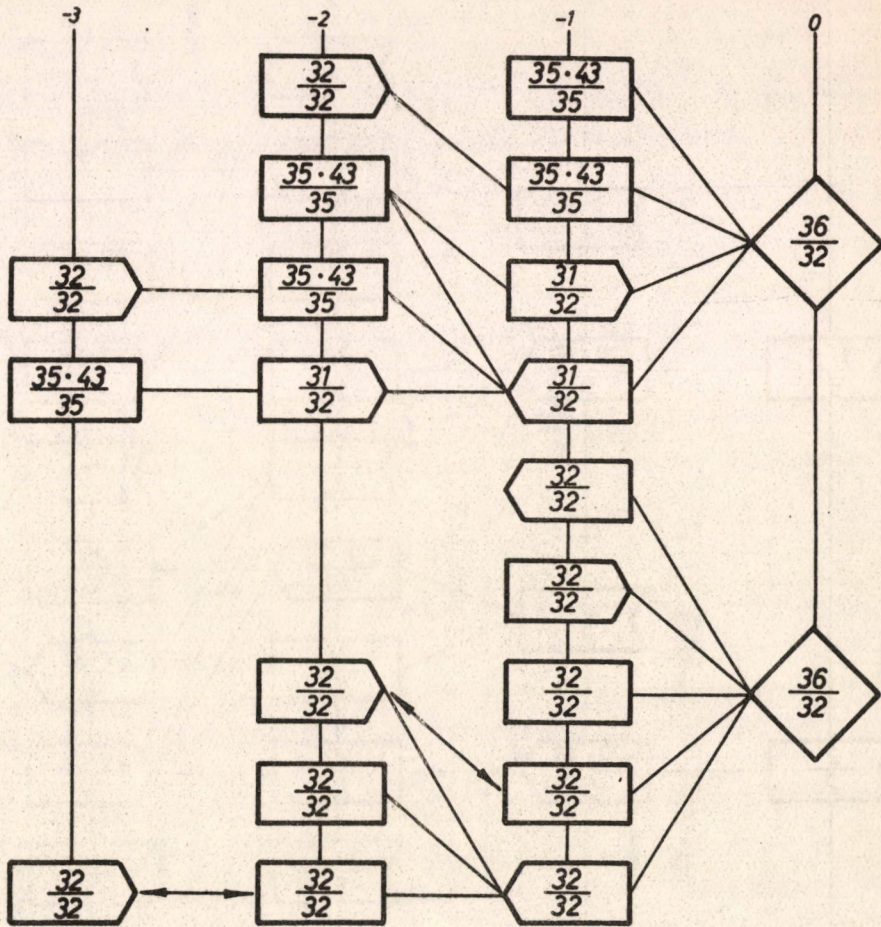


Fig. 12.

$N_{37}$

A

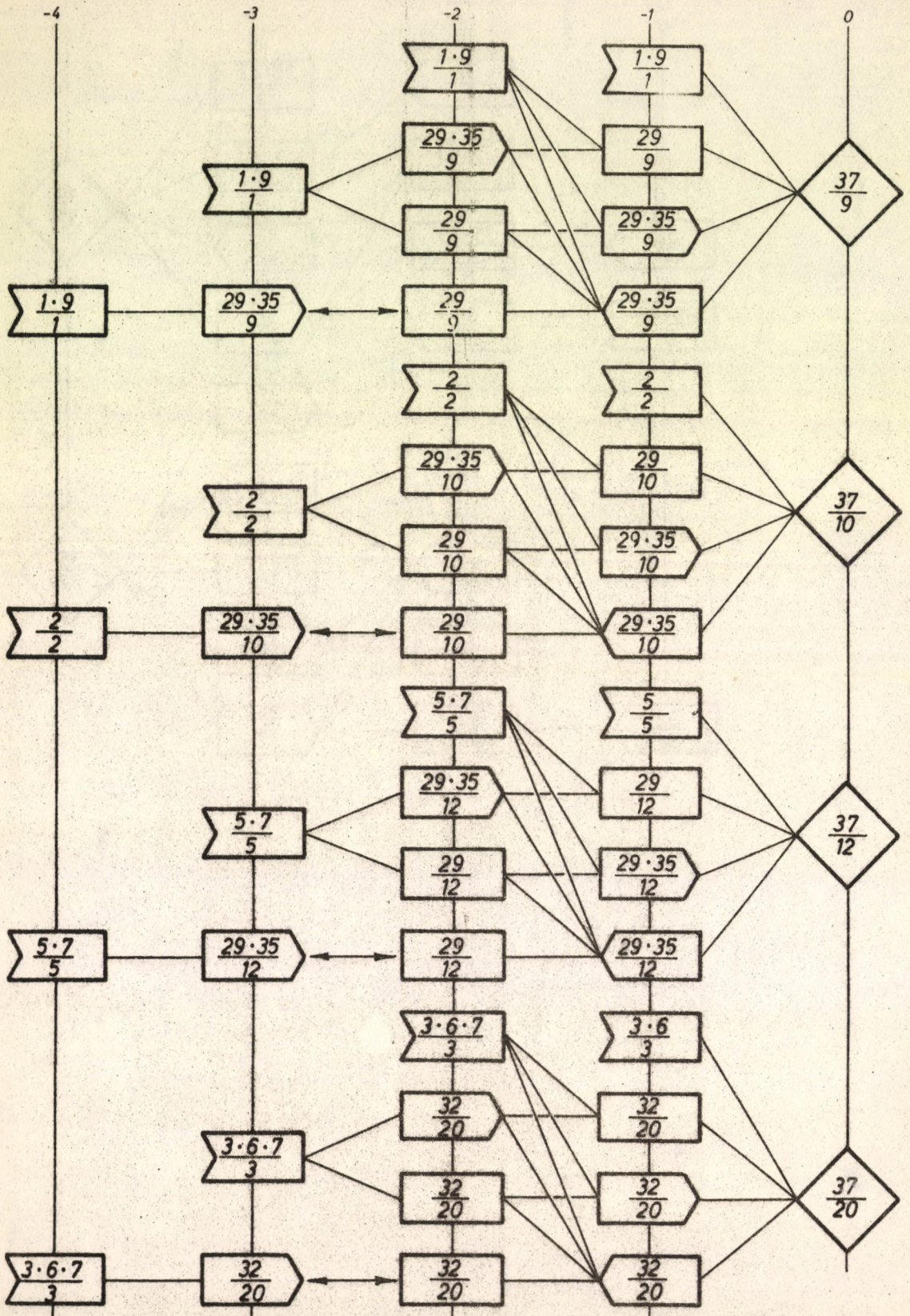




Fig. 12.

$N_{37}$

B

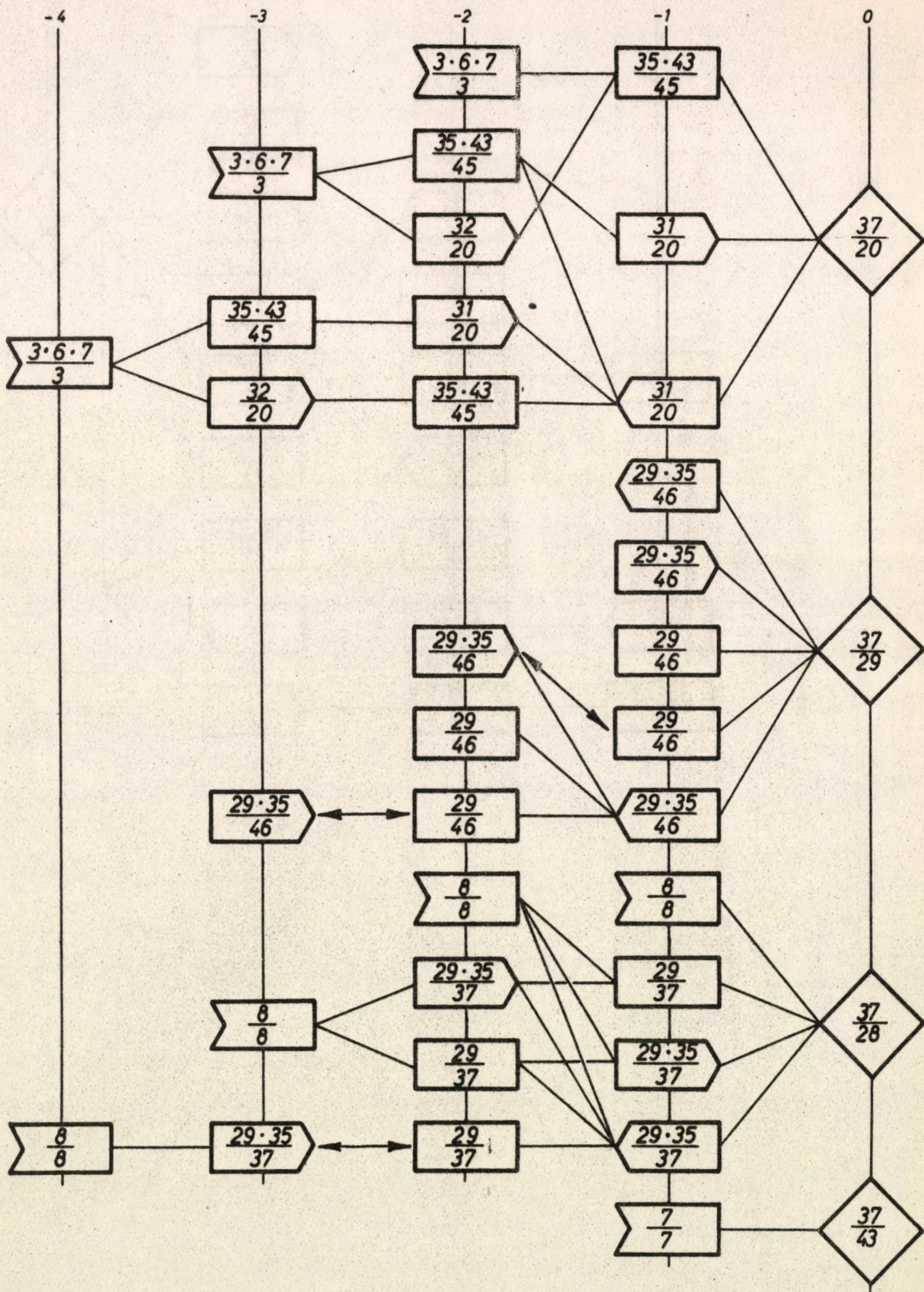


Fig. 12.

$N_{37}$

C

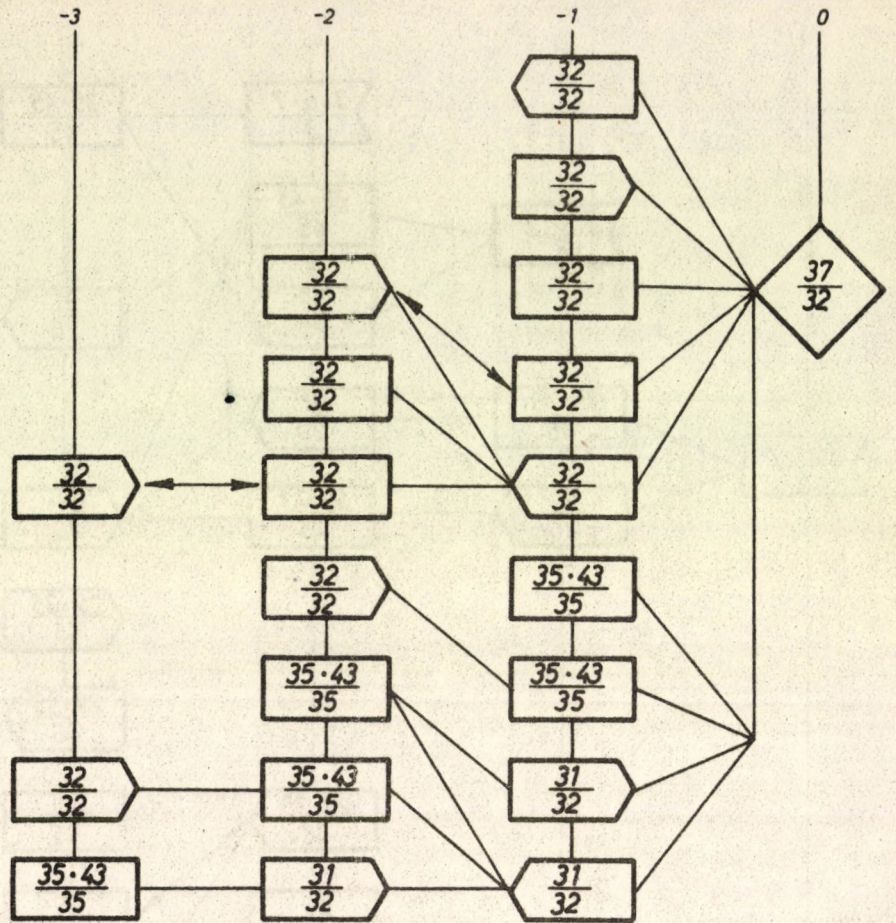


Fig. 13.

$N_{39}$

A

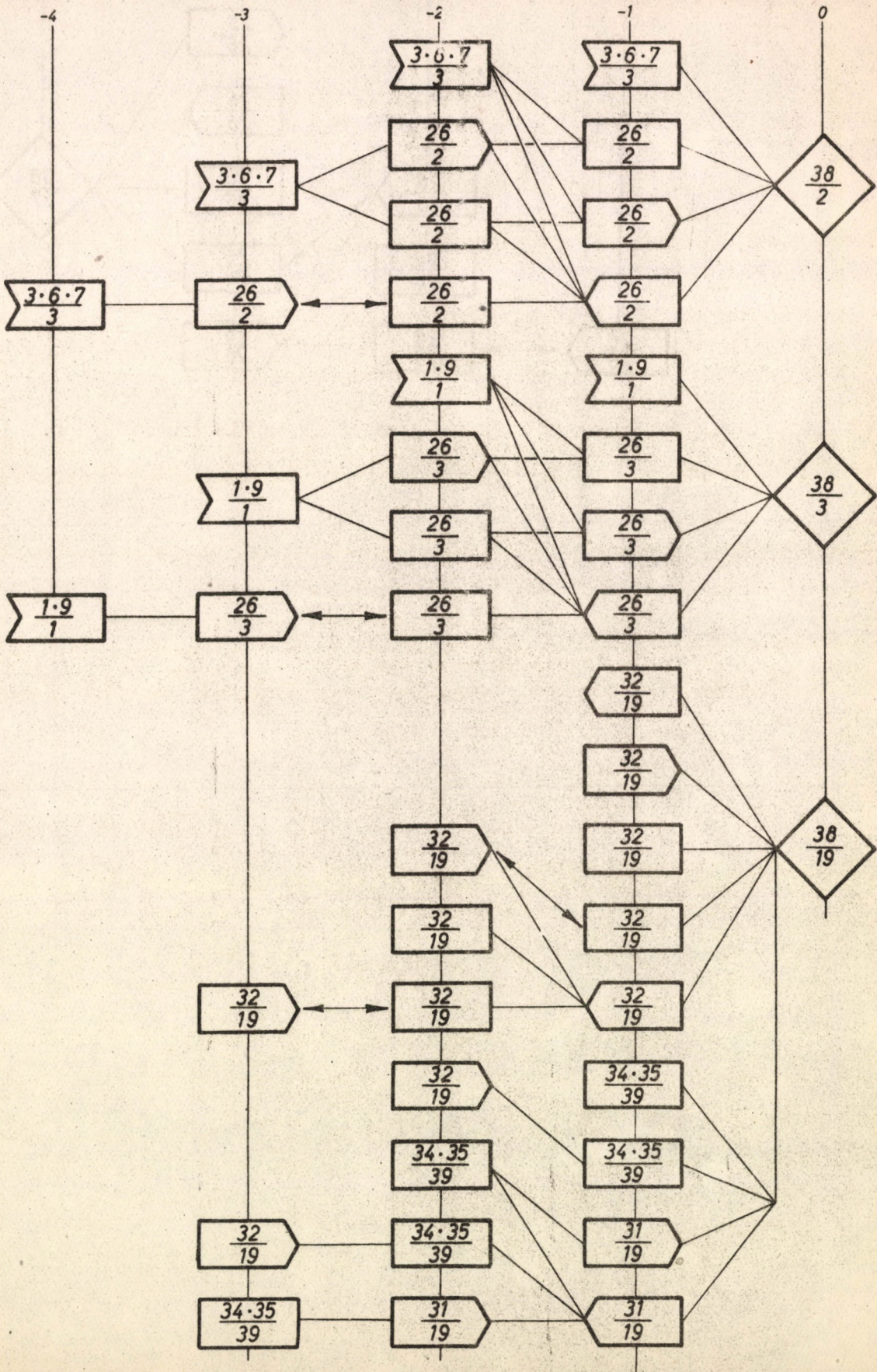


Fig. 13.

$N_{30}$

B

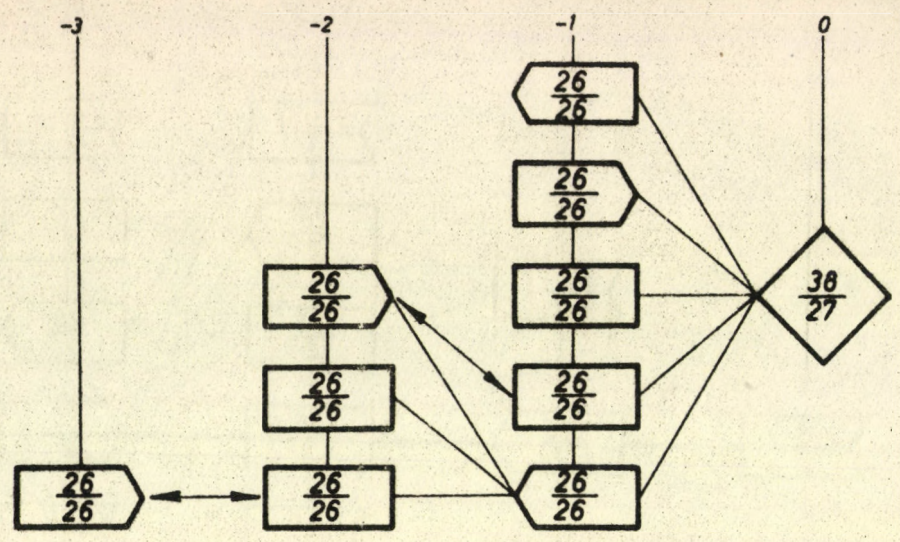


Fig. 14.

$N_{35}$

4

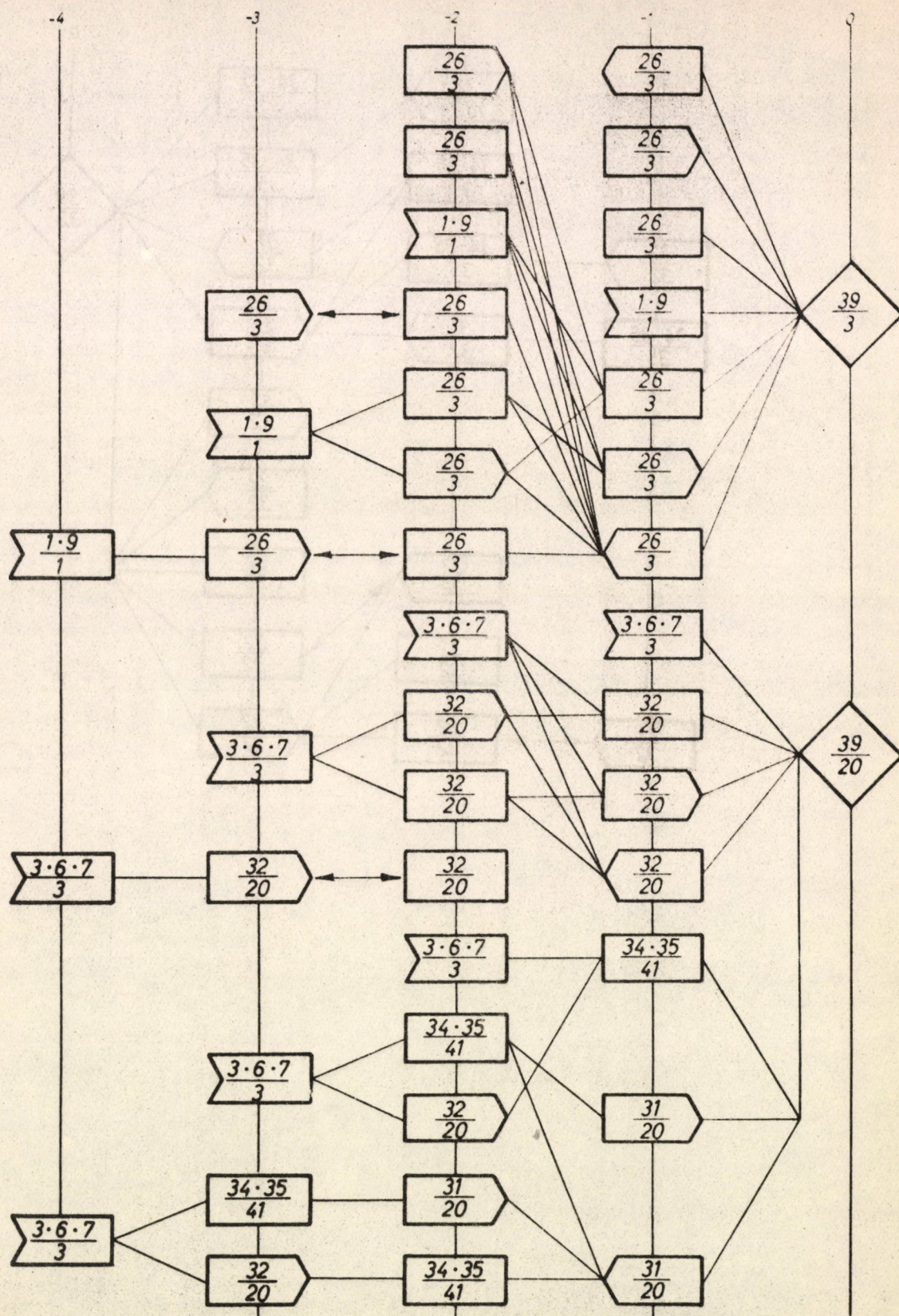


Fig. 14.

$N_{39}$

B

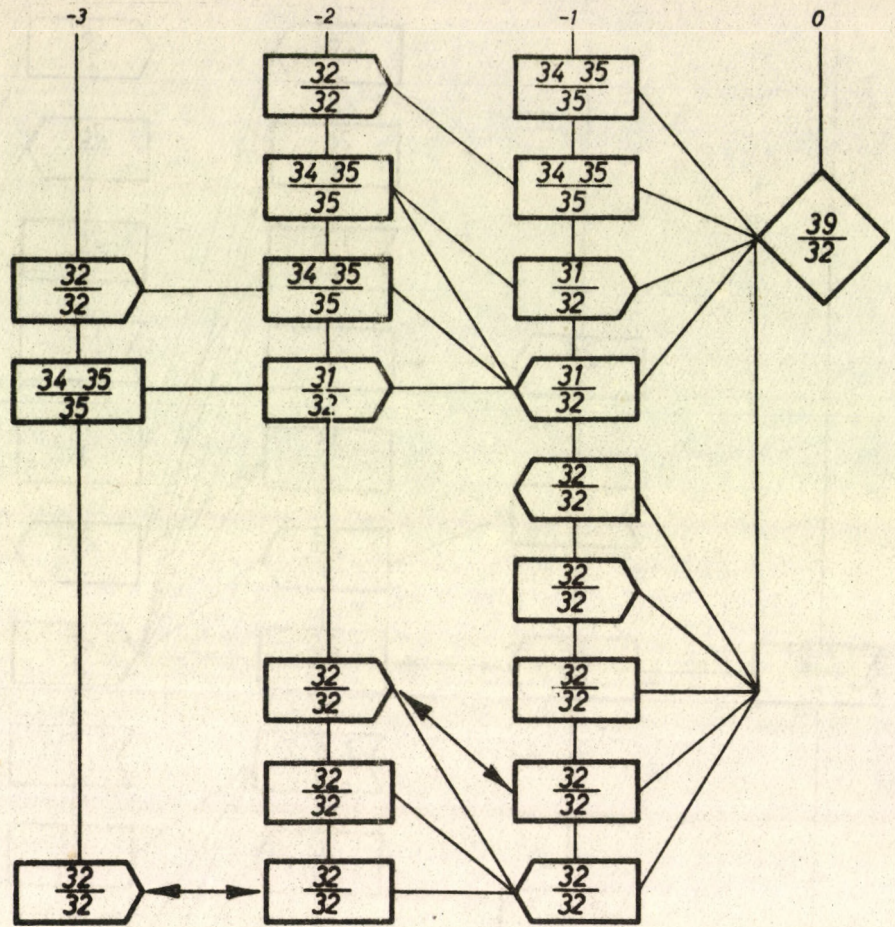


Fig. 15.

N<sub>40</sub>

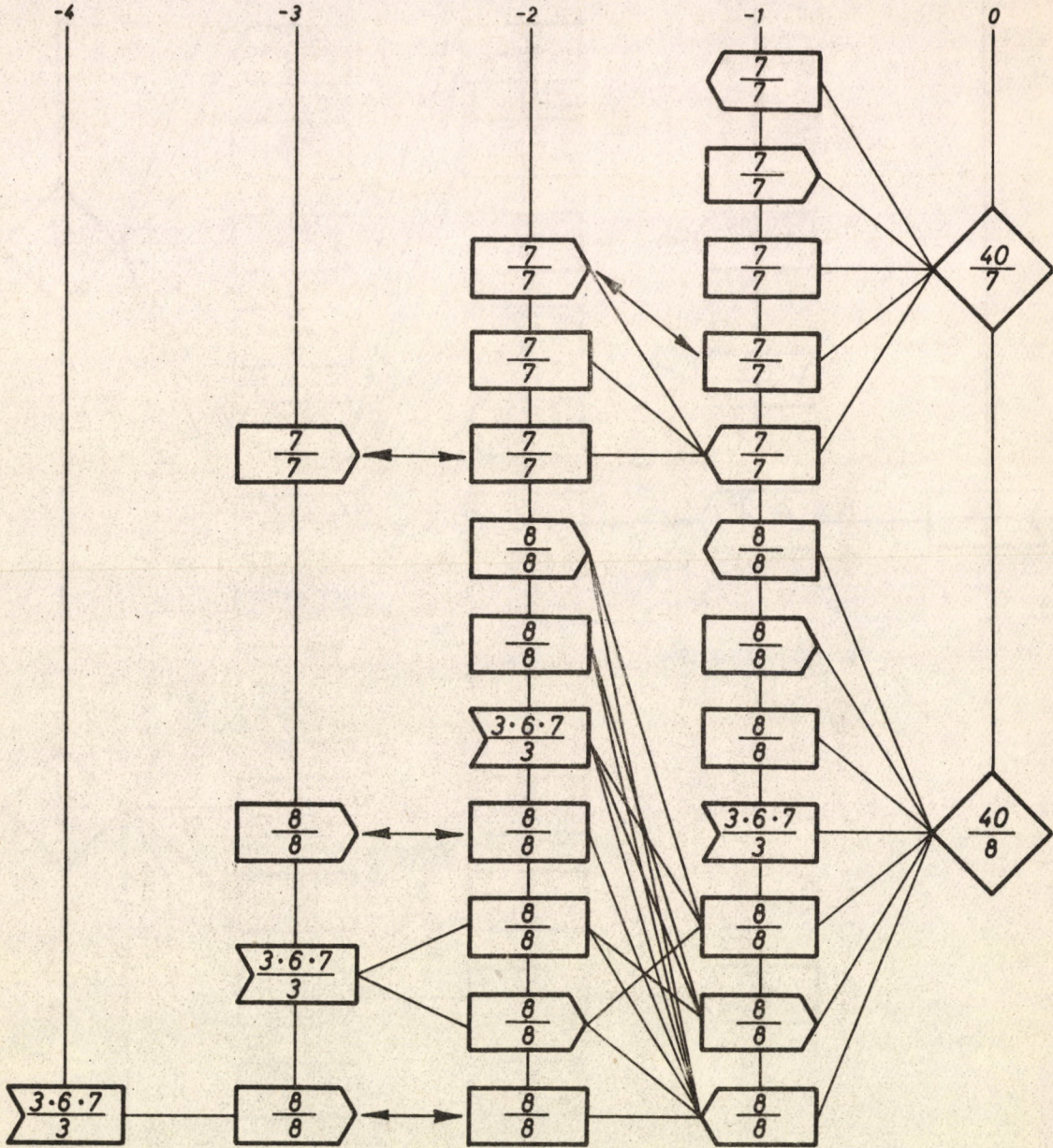
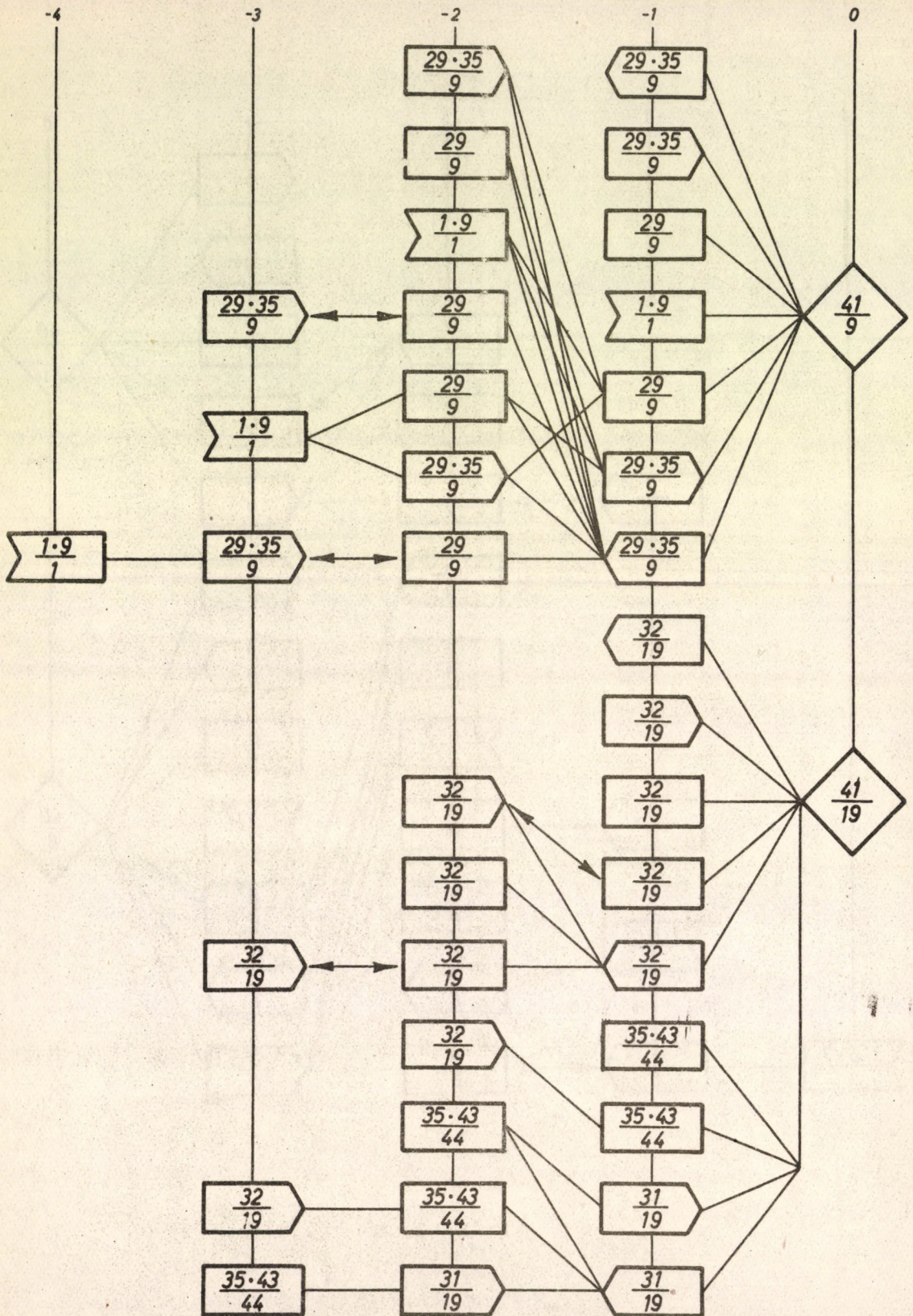


Fig. 16.

$N_{41}$





#### 4. Formulae for the Programs

The programs of machine translation are based on the train of thought of an average translator. As by using the system of so called "opening" and "final" words it is possible for the computer to analyse not only separate words but whole word groups as well, the programs for the homonym nominal forms - as it has already been mentioned - can be drawn up with the help of the connections by means of which the translator decides the role of a given form.

The conditions of deciding in which case the homonym noun form actually is - necessary for both the translator and the computer - can be received by taking out the members of the word groups that show the exact role of the homonym form given in the group, or at least reduce the number of the possible variants. The connections containing these conditions - completed with the list of possibilities not giving any result - can be made easy of access for the computer by using formulas.

In the formulas given below the symbols of formal logic are used. The symbols express the lack (  $\overline{\quad}$  ) of a substantive or preposition, a change of choice (  $+ = \text{or}$  ), a connection (  $\cdot = \text{and}$  ) and an outcome (  $\longrightarrow = \text{then}$  ).

The conditions by means of which the role of a given homonym noun form can be decided are the following:

N<sub>25</sub>: The case of the noun can be decided only if in the given word group there is a preposition governing accusative, accusative and instrumental or accusative and prepositional cases. This fact can be expressed with formulae like this:

$$(\text{Pr}_3 + \text{Pr}_6 + \text{Pr}_7) \longrightarrow \text{N}_2 \quad (1)$$

$$\overline{\text{Pr}} \longrightarrow \text{N}_{25} \longrightarrow \text{Q}_{25}^{1/} \quad (2)$$

---

1/ Q = further analysis is necessary.

N<sub>26</sub>:

As there is no noun in prepositional case without a preposition, if there is no preposition in the word group or the preposition governs only dative case, the noun is in the dative case.

If there is a preposition governing only prepositional case or one governing accusative and prepositional cases, the noun is in the prepositional case.

If the preposition governs both dative and prepositional case, the decision can be made only with the help of other substantives (A, Nu, P) in the word group. With formulas:

$$\left\{ \overline{\text{Pr}} + \text{Pr}_2 + \left[ \text{Pr}_8 \cdot (\text{A}_{27} + \text{Nu}_{27} + \text{P}_{27}) \right] \right\} \rightarrow \text{N}_4 \quad (3)$$

$$\left\{ \text{Pr}_5 + \text{Pr}_7 + \text{Pr}_8 \cdot (\text{A}_{28} + \text{Nu}_{28} + \text{P}_{28}) \right\} \rightarrow \text{N}_6 \quad (4)$$

$$\left[ \overline{\text{Pr}_8 \cdot (\text{A}_{27} + \text{A}_{28} + \text{Nu}_{27} + \text{Nu}_{28} + \text{P}_{27} + \text{P}_{28})} \right] \rightarrow$$

$$\rightarrow \text{N}_{26} \rightarrow \text{Q}_{26} \quad (5)$$

N<sub>27</sub>:

If the word group contains a preposition governing accusative case or one governing accusative and instrumental or accusative and prepositional cases, the noun is in the accusative case.

If there is a preposition governing genitive or one governing genitive and instrumental cases, the noun is in the genitive case.

If there is no preposition but there is a numeral governing genitive case, the changed code number denotes nominative case of the plural. With formulae:

$$(\text{Pr}_3 + \text{Pr}_6 + \text{Pr}_7) \rightarrow \text{N}_2 \quad (6)$$

$$(\text{Pr}_1 + \text{Pr}_9) \rightarrow \text{N}_3 \quad (7)$$

$$\left[ \overline{\text{Pr}} \cdot (\text{Nu}_{34} + \text{Nu}_{35}) \right] \rightarrow \text{N}_{19} \quad (8)$$

$$\left[ \overline{\text{Pr}} \cdot (\text{Nu}_{34} + \text{Nu}_{35}) \right] \rightarrow \text{N}_{27} \rightarrow \text{Q}_{27} \quad (9)$$

N<sub>28</sub>: The conditions are the same as with N<sub>26</sub>. But as the preposition "по" practically never occurs in prepositional case with a noun of feminine gender, it makes the decision to be prepositional case impossible just for Pr<sub>2</sub>. With formulas:

$$(\overline{\text{Pr}} + \text{Pr}_2 + \text{Pr}_8) \rightarrow \text{N}_{10} \quad (10)$$

$$(\text{Pr}_5 + \text{Pr}_7) \rightarrow \text{N}_{12} \quad (11)$$

N<sub>30</sub>: The conditions are the same as with N<sub>25</sub>. With formulas:

$$(\text{Pr}_3 + \text{Pr}_6 + \text{Pr}_7) \rightarrow \text{N}_{14} \quad (12)$$

$$\overline{\text{Pr}} \rightarrow \text{N}_{30} \rightarrow \text{Q}_{30} \quad (13)$$

N<sub>31</sub>: If the word group contains a preposition governing only accusative case or one governing accusative and instrumental cases, the noun is in the accusative case.

If there is a preposition governing only prepositional case or one governing dative and prepositional cases, the noun is in the prepositional case.

If there is a preposition governing only prepositional case or one governing dative and prepositional cases, the noun is in the prepositional case.

If there is a preposition governing accusative and prepositional cases, the decision can be made only with the help of other substantives (A, Nu, P) in the word group.

If there is no preposition, the noun can be either in the nominative or in the accusative case. With formulae:

$$\left\{ \text{Pr}_3 + \text{Pr}_6 + \left[ \text{Pr}_7 \cdot (\text{A}_{30} + \text{Nu}_{30} + \text{P}_{30}) \right] \right\} \rightarrow \text{N}_{14} \quad (14)$$

$$\left\{ \text{Pr}_5 + \text{Pr}_8 + \left[ \text{Pr}_7 \cdot (\text{A}_{28} + \text{Nu}_{28} + \text{P}_{28}) \right] \right\} \rightarrow \text{N}_{18} \quad (15)$$

$$\left[ \text{Pr}_7 \cdot (\text{A}_{28} + \text{A}_{30} + \text{Nu}_{28} + \text{Nu}_{30} + \text{P}_{28} + \text{P}_{30}) \right] \rightarrow \text{N}_{42} \rightarrow \text{Q}_{42} \quad (16)$$

$$\overline{\text{Pr}} \rightarrow \text{N}_{30} \rightarrow \text{Q}_{30} \quad (17)$$

N<sub>32</sub>: The conditions are the same as with N<sub>25</sub>. With formulae:

$$(\text{Pr}_3 + \text{Pr}_6 + \text{Pr}_7) \rightarrow \text{N}_{20} \quad (18)$$

$$\overline{\text{Pr}} \rightarrow \text{N}_{32} \rightarrow \text{Q}_{32} \quad (19)$$

N<sub>33</sub>: If the word group contains a preposition governing only accusative case or one governing accusative and prepositional cases, the noun is in the accusative case.

If there is a preposition governing only genitive case or one governing genitive and instrumental cases, the noun is in the genitive case. The noun is in the genitive case also in the word groups without any preposition but containing a numeral of the "пяти, сорока", etc. type (Nu<sub>48</sub>).

If there is no preposition but there is a numeral of the "пять, сто" etc. type ( $Nu_{35}$ ) and a pronoun in the nominative or accusative of the plural, the changed code number of the noun will denote nominative case of the plural.

If before the numeral there is a pronoun form indicating accusative, genitive and prepositional cases of the plural, the changed code number of the noun will denote accusative case of the plural.

Finally, if the numeral takes the first place of the word group, the noun will get a code number for both nominative and accusative case of the plural. With formulae:

$$(Pr_3 + Pr_6 + Pr_7) \rightarrow N_{20} \quad (20)$$

$$[Pr_7 + Pr_9 + (\overline{Pr} \cdot Nu_{48})] \rightarrow N_{21} \quad (21)$$

$$[\overline{Pr} \cdot (P_{32} \cdot Nu_{35})] \rightarrow N_{19} \quad (22)$$

$$[\overline{Pr} \cdot (P_{31} \cdot Nu_{35} \cdot P_{32})] \rightarrow N_{20}^{1/} \quad (23)$$

$$[\overline{Pr} \cdot (Nu_{35} \cdot P_{31} \cdot P_{32})] \rightarrow N_{32} \rightarrow Q_{32}^{1/} \quad (24)$$

$$[\overline{Pr} \cdot (Nu_{35} + Nu_{48})] \rightarrow N_{33} \rightarrow Q_{33} \quad (25)$$

N<sub>34</sub>: If the word group contains a preposition governing only genitive case or one governing both genitive and instrumental cases, or there is no preposition but there is an adjective in the genitive case of the singular, or a numeral in the same case or a pronoun in the same case, the noun is in the genitive case of the singular.

1/  $\longrightarrow$  = the order of succession cannot be changed!

If there is a preposition governing only accusative case or one governing accusative and instrumental case, or one governing accusative and prepositional cases, the noun is in the accusative case of the plural.

If there is no preposition but there is a numeral governing genitive case of the singular, the changed code number of the noun denotes nominative and accusative cases of the plural. The same is the code number if there is neither a preposition nor a numeral governing genitive case of the singular but there is an adjective, numeral or pronoun (with a code number denoting nominative and accusative cases of the plural). With formulae:

$$\left\{ \text{Pr}_1 + \text{Pr}_9 + \left[ \overline{\text{Pr}} \cdot (\text{A}_{26} + \text{Nu}_{26} + \text{P}_{26}) \right] \right\} \rightarrow \text{N}_{15} \quad (26)$$

$$(\text{Pr}_3 + \text{Pr}_6 + \text{Pr}_7) \rightarrow \text{N}_{20} \quad (27)$$

$$\left\{ \left[ \overline{\text{Pr}} \cdot (\text{Nu}_{34} + \text{Nu}_{35}) \right] + \left[ \overline{\text{Pr}} \cdot (\text{Nu}_{34} + \text{Nu}_{35}) \cdot (\text{A}_{32} + \text{Nu}_{32} + \text{P}_{32}) \right] \right\} \rightarrow \text{N}_{32} \rightarrow \text{Q}_{32} \quad (28)$$

N<sub>35</sub>: If the word group contains a preposition governing only prepositional case or one governing accusative and prepositional cases<sup>1/</sup> or one governing dative and prepositional case the noun is in the prepositional case of the singular.

<sup>1/</sup> As this group of nouns contains - for the most part - concrete nouns the plural of which is not used (the other members of the group are not likely to occur in technical texts), in this program with a preposition governing accusative and prepositional cases the noun can be taken for one being in the prepositional case of the singular.

If there is a preposition governing only accusative case or one governing accusative and instrumental cases, the noun is in the accusative case of the plural.

If there is no preposition, the changed code number of the noun will denote both nominative and accusative cases of the plural. With formulae:

$$(\text{Pr}_5 + \text{Pr}_7 + \text{Pr}_8) \rightarrow N_6 \quad (29)$$

$$(\text{Pr}_3 + \text{Pr}_6) \rightarrow N_{20} \quad (30)$$

$$\overline{\text{Pr}} \rightarrow N_{32} \rightarrow Q_{32} \quad (31)$$

N<sub>36</sub>: If the word group contains a preposition governing only genitive case or one governing genitive and instrumental cases or there is no preposition but there is an adjective in the genitive case of the singular (feminine) or a numeral in the same case or a pronoun in the same case, the noun is in the genitive case of the singular.

If there is a preposition governing only accusative case or one governing either accusative and instrumental or accusative and prepositional case, the changed code number of the noun denotes accusative case of the plural.

If there is no preposition but there is a numeral governing genitive case of the singular, or if there is neither a preposition nor a numeral governing genitive case of the singular but there is an adjective with a code number for nominative and accusative cases of the plural or a numeral with the same code number or a pronoun with the same code number, the noun can be either in the nominative or accusative case of the plural.

With formulae:

$$\left\{ \text{Pr}_1 + \text{Pr}_9 + \left[ \overline{\text{Pr}} \cdot (\text{A}_{29} + \text{Nu}_{29} + \text{P}_{29}) \right] \right\} \rightarrow \text{N}_9 \quad (32)$$

$$(\text{Pr}_3 + \text{Pr}_6 + \text{Pr}_7) \rightarrow \text{N}_{20} \quad (33)$$

$$\left\{ \left[ \overline{\text{Pr}} \cdot (\text{Nu}_{35} + \text{Nu}_{43}) \right] + \left[ \overline{\text{Pr}} \cdot (\text{Nu}_{35} + \text{Nu}_{43}) \cdot (\text{A}_{32} + \text{Nu}_{32} + \text{P}_{32}) \right] \right\} \rightarrow \text{N}_{32} \rightarrow \text{Q}_{32} \quad (34)$$

N<sub>37</sub>: If the word group contains a preposition governing only genitive case or one governing genitive and instrumental case, the noun is in the genitive case of the singular.

If there is a preposition governing only dative case, the noun is in the dative case of the singular.

The noun is in the prepositional case of the singular if the word group contains a preposition governing only prepositional case or one governing accusative and prepositional case and an adjective with a code number for genitive, dative, instrumental and prepositional cases of the singular or either a numeral or a pronoun with the same code number at the same time.

If there is a preposition governing only accusative case or one governing accusative and instrumental case or there is a preposition governing accusative and prepositional cases and an adjective with a code number for nominative and accusative cases of the plural or either a numeral or a pronoun with the same code number, the noun is in the accusative case of the plural.

If there is no preposition but the word group contains either an adjective with a code number for genitive, dative, instrumental and prepositional cases of the sin-



gular for a numeral with the same code number or a pronoun with the same code number, the noun gets a code number denoting genitive and dative cases of the singular.

If there is a preposition governing accusative and prepositional case but there is neither an adjective with a code number for both genitive, dative, instrumental and prepositional cases of the singular and nominative and accusative of the plural nor a numeral with the same code number nor a pronoun with the same code number nor a numeral governing genitive case of the singular, the noun can be either in the prepositional case of the singular, or in the accusative case of the plural.

If there is no preposition but there is a numeral governing genitive case of the singular, the changed code number of the noun will indicate nominative and accusative cases of the plural. The noun is given the same code number if the word group does not contain any numeral governing genitive case of the singular but there is an adjective, numeral or pronoun (with a code number for nominative and accusative of the plural) in it. With formulae:

$$\therefore (Pr_1 + Pr_9) \rightarrow N_9 \quad (35)$$

$$Pr_2 \rightarrow N_{10} \quad (36)$$

$$\left\{ Pr_5 + \left[ Pr_7 \cdot (A_{29} + Nu_{29} + P_{29}) \right] \right\} \rightarrow N_{32} \rightarrow Q_{32} \quad (37)$$

$$\left\{ Pr_3 + Pr_6 + \left[ Pr_7 \cdot (A_{32} + Nu_{32} + P_{32} + Nu_{35} + Nu_{43}) \right] \right\} \rightarrow N_{20} \quad (38)$$

$$\left[ Pr \cdot (A_{29} + A_{35} + Nu_{29} + P_{29} + P_{35}) \right] \rightarrow N_{29} \rightarrow Q_{29} \quad (39)$$

$$Pr_8 \rightarrow N_{28} \rightarrow Q_{28} \quad (40)$$

$$\left[ \text{Pr}_7 \cdot (\overline{A_{29} + A_{32} + \text{Nu}_{29} + \text{Nu}_{32} P_{29} + P_{32} + \text{Nu}_{35} + \text{Nu}_{43}}) \right] \rightarrow$$

$$\rightarrow N_{43} \rightarrow Q_{43} \quad (41)$$

$$\left\{ \left[ \overline{\text{Pr}} \cdot (\text{Nu}_{35} + \text{Nu}_{43}) \right] + \left[ \overline{\text{Pr}} \cdot (\overline{\text{Nu}_{35} + \text{Nu}_{43}}) \right] \cdot \right.$$

$$\left. \cdot (A_{32} + \text{Nu}_{32} + P_{32}) \right\} \rightarrow N_{32} \rightarrow Q_{32} \quad (42)$$

N<sub>38</sub>: If the word group contains a preposition governing only accusative case or one governing either accusative and instrumental cases or accusative and prepositional cases, the noun is in the accusative case of the singular.

If there is a preposition governing either only genitive case or genitive and instrumental cases, the noun is in the genitive case of the singular.

If there is no preposition but there is a numeral governing genitive case of the singular, or there is no preposition, no numeral governing genitive case of the singular but there is an adjective with a code number for nominative and accusative of the plural or either a numeral or a pronoun with the same code number, the noun is in the nominative case of the plural.

If there is no preposition, no numeral governing genitive case of the singular but there is an adjective with a code number for accusative and genitive cases of the singular, or either a numeral or a pronoun with the same code number, the changed code number of the noun denotes accusative and genitive cases of the singular. With formulae:

$$(\text{Pr}_3 + \text{Pr}_6 + \text{Pr}_7) \rightarrow N_2 \quad (43)$$

$$(\text{Pr}_1 + \text{Pr}_9) \rightarrow N_3 \quad (44)$$

$$\left\{ \left[ \overline{\text{Pr}} \cdot (\text{Nu}_{34} + \text{Nu}_{35}) \right] + \left[ \overline{\text{Pr}} \cdot (\text{Nu}_{34} + \text{Nu}_{35}) \cdot (A_{32} + \text{Nu}_{32} + P_{32}) \right] \right\} \rightarrow N_{19} \quad (45)$$

$$\left[ \overline{\text{Pr}} \cdot (\text{Nu}_{34} + \text{Nu}_{35}) \cdot (A_{26} + \text{Nu}_{26} + P_{26}) \right] \rightarrow N_{27} \rightarrow Q_{27} \quad (46)$$

N<sub>39</sub>: If the word group contains a preposition governing only genitive case or one governing genitive and instrumental cases, besides, if there is no preposition in the word group but there is an adjective with a code number for accusative and genitive cases of the singular or either a numeral or a pronoun with the same code number, the noun is in the genitive case of the singular.

If there is a preposition governing only accusative case or one governing either accusative and instrumental or accusative and prepositional cases, the noun is in the accusative case of the plural.

If there is no preposition but there is a numeral governing genitive case of the singular, the changed code number of the noun indicates nominative and accusative cases of the plural. The noun is given the same code number if there is neither a preposition nor a numeral governing genitive case of the singular, but there is an adjective, numeral or pronoun (with a code number for nominative and accusative cases of the plural).

With formulae:

$$\left\{ \text{Pr}_1 + \text{Pr}_9 + \left[ \overline{\text{Pr}} \cdot (A_{26} + \text{Nu}_{26} + P_{26}) \right] \right\} \rightarrow N_3 \quad (47)$$

$$(\text{Pr}_3 + \text{Pr}_6 + \text{Pr}_7) \rightarrow N_{20} \quad (48)$$

$$\left[ \overline{\text{Pr}} \cdot (\text{Nu}_{34} + \text{Nu}_{35}) \right] + \left[ \overline{\text{Pr}} \cdot (\text{Nu}_{34} + \text{Nu}_{35}) \cdot (A_{32} + \text{Nu}_{32} + P_{32}) \right] \rightarrow N_{32} \rightarrow Q_{32} \quad (49)$$

N<sub>40</sub>: If there is no preposition in the word group but there is an adjective in the nominative case of the singular (feminine) or either a numeral or a pronoun in the same case, the noun is in the nominative case of the singular.

If there is a preposition governing only accusative case or one governing either accusative and instrumental or accusative and prepositional cases, besides, if there is no preposition but there is an adjective in the accusative case of the singular (feminine) or either a numeral or a pronoun in the same case, the noun is in the accusative case of the singular. With formulae:

$$\left[ \overline{\text{Pr}} \cdot (A_7 + \text{Nu}_7 + P_7) \right] \rightarrow N_7 \quad (50)$$

$$\left\{ \text{Pr}_3 + \text{Pr}_6 + \text{Pr}_7 + \left[ \overline{\text{Pr}} \cdot (A_8 + \text{Nu}_8 + P_8) \right] \right\} \rightarrow N_8 \quad (51)$$

N<sub>41</sub>: If the word group contains a preposition governing only genitive case or one governing genitive and instrumental cases, besides, if there is no preposition but there is an adjective with a code number for genitive, dative, instrumental and prepositional of the singular (feminine) or a numeral with the same code number or a pronoun with the same code number, the noun is in the genitive case of the singular.

If there is no preposition but there is a numeral governing genitive case of the singular or there is no preposition but there is an adjective with a code number for nominative and accusative cases of the plural or a numeral with the same code number or a pronoun with the same code number, the noun is in the nominative case of the plural. With formulae:

$$\left\{ \text{Pr}_1 + \text{Pr}_9 + \left[ \overline{\text{Pr}} \cdot (\text{A}_{29} + \text{A}_{35} + \text{Nu}_{29} + \text{P}_{29} + \text{P}_{35}) \right] \right\} \rightarrow \text{N}_9 \quad (52)$$

$$\left\{ \left[ \overline{\text{Pr}} \cdot (\text{Nu}_{35} + \text{Nu}_{43}) + \overline{\text{Pr}} \cdot (\overline{\text{Nu}_{35} + \text{Nu}_{43}}) \cdot (\text{A}_{32} + \text{Nu}_{32} + \text{P}_{32}) \right] \right\} \rightarrow \text{N}_{19} \rightarrow \text{Q}_{19} \quad (53)$$

### 5. The Programs

The process necessary for the computer to perform the code changing of the homonym nominal forms consists of two parts:

a) First the role of the homonym noun form in the word group is to be cleared. The computer is given the instructions enabling it to do this work in programs based on the conditions established in the formulae of point 1. This programs can be named Determining Programs. As the conditions are different for each of the homonym noun forms, there are as many determining programs as there are homonym noun forms (D.I. - D.XVL).

b) The case of the noun having been determined the code numbers of the noun and other substantives of the word group are to be changed (see: point 2). The programs containing the instructions needed for this work can be called Changing Programs. As the changing of the code numbers can be perfor-

med by means of identical operations in the various word groups belonging to the same type, the use of this changing programs makes the system of instructions much shorter as if the instructions for changing the code numbers were given together with the case determining programs of the homonym noun forms. E.g.: in the case of N<sub>37</sub> the process of analysis would contain 137 "steps" without using changing programs, while using this programs the same word can be done in 56 "steps" maximum.

The character of a changing program depends on the possibilities given by the members of a word group in determining the role of the noun form. According to these possibilities there are 6 programs (C.I. - C.VI.) necessary to perform the actual changing of the code numbers of the substantives and prepositions.

- C.I. : for the type of word groups where the result of the analysis depends on a preposition only.
- C.II. : for the type where the result depends on a preposition and another substantive (A, Nu, P).
- C.III.: for the type where the result depends on a substantive (A, Nu, P) only.
- C.IV. : for the type where the result depends on a preposition and numeral governing genitive case.
- C.V. : for the type where the result depends on a numeral governing genitive case.
- C.VI. : for the type where the result depends on numeral governing genitive case and a pronoun.

The instructions of the single changing program can be used for changing the code number of not only one homonym noun form, but those of several homonym noun forms (and - at

the same time - those of the other substantives and prepositions forming one word group with them). This is possible only if the symbols given in the instructions of a changing program are always replaced with numerical reviews (code numbers) corresponding to the word group being actually analysed. The numbers with which the symbols of the changing programs must be replaced are given at the end of the description of each determining program.

When the word group does not contain any preposition or other substantives by means of which the role of the homonym noun form could be cleared or the analysis of the word group gives only a "partial solution"<sup>1/</sup> further analysis is necessary. As in these cases the decision cannot be made without having semantic informations, the programs for this analysis (in the following "Q" programs) can be made up only after working out the semantic code system.

#### a) Determining Programs

The steps contained in the determining programs can be put down as follows:

##### D.I. (N<sub>25</sub>; fig. 1.; form. 1-2.)

1. (2, 3) Is there: Pr<sub>3</sub> in the word group?
2. Turn to C. I.
3. (2, 4) Is there: Pr<sub>6</sub>?
4. (2, 5) Is there: Pr<sub>7</sub>?
5. Turn to Q<sub>25</sub>

---

<sup>1/</sup> By "partial solution" it is meant that the role of a given homonym noun form is not quite cleared but the possible variants are reduced to two.

Values of the parameters in the changing program:

<u>Input:</u>		<u>Output:</u>	
<u>Value:</u>	<u>Sign:</u>	<u>Sign:</u>	<u>Value:</u>
$Pr_3 + Pr_6 + Pr_7^{1/}$	= $Pr_i$	$Pr_i,$	$\Rightarrow Pr_3$
$A_{25} + A_{35}$	= $A_i$	$A_i,$	$\Rightarrow A_2$
$Nu_{25}$	= $Nu_i$	$Nu_i,$	$\Rightarrow Nu_2$
$P_{25} + P_{35}$	= $P_i$	$P_i,$	$\Rightarrow P_2$
$N_{25}$	= $N_i$	$N_i,$	$\Rightarrow N_2$

D. II. ( $N_{26}$ ; fig. 2.; form. 3-5.)

1. (2, 4) Is there:  $Pr_2$  in the word group?
2. (3, -) Take:  $k, k'.^{2/}$
3. Turn to C. I.
4. (5, 7) Is there:  $Pr_5$ ?
5. (6, -) Take:  $m, m'.$
6. Turn to C. I.

---

1/ + = or.

2/ This step - an all the other steps of this kind - give the signs with which the signs of the changing programs must be replaced. The computer turns to the changing program, performs the replacements given in the instructions of the determining program and choosing the value correspondent to the sign got this way among the output values given at the end of the determining program, gets the changed code number.



7. (5, 8) Is there:  $Pr_7$ ?
8. (9,14) Is there:  $Pr_8$ ?
9. (10,11) Is there: A ?
10. Turn to C.II.
11. (10,12) Is there: Nu?
12. (10,13) Is there: P ?
13. Turn to  $Q_{26}$ .
14. (17,15) Is there: A ?
15. (17,16) Is there: Nu?
16. (17,19) Is there: P ?
17. (18, -) Take:  $k, k'$ .
18. Turn to C.III.
19. Turn to  $Q_{26}$ .

Values of the parameters in the changing programs:

<u>Input:</u>		<u>Output:</u>	
<u>Value:</u>	<u>Sign:</u>	<u>Sign:</u>	<u>Value:</u>
$Pr_2$	= $Pr_k$	$Pr_k,$	$\Rightarrow Pr_2$
$Pr_5 + Pr_7$	= $Pr_m$	$Pr_m,$	$\Rightarrow Pr_5$
$A_{27}$	= $A_k$	$A_k,$	$\Rightarrow A_4$
$Nu_{27}$	= $Nu_k$	$Nu_k,$	$\Rightarrow Nu_4$
$P_{27}$	= $P_k$	$P_k,$	$\Rightarrow P_4$
$A_{28}$	= $A_m$	$A_m,$	$\Rightarrow A_6$

Nu <sub>28</sub>	=	Nu <sub>m</sub>	Nu <sub>m</sub> ,	=>	Nu <sub>6</sub>
P <sub>28</sub>	=	P <sub>m</sub>	P <sub>m</sub> ,	=>	P <sub>6</sub>
N <sub>26</sub>	=	N <sub>k</sub>	N <sub>k</sub> ,	=>	N <sub>4</sub>
N <sub>26</sub>	=	N <sub>m</sub>	N <sub>m</sub> ,	=>	N <sub>6</sub>

D. III. (N<sub>27</sub>; fig. 3.; form. 6-9)

1. (2, 4) Is there: Pr<sub>3</sub> in the word group?
2. (3, -) Take: k, k'.
3. Turn to C. I.
4. (2, 5) Is there: Pr<sub>6</sub>?
5. (2, 6) Is there: Pr<sub>7</sub>?
6. (7, 9) Is there: Pr<sub>1</sub>?
7. (8, -) Take: m, m'.
8. Turn to C. I.
9. (7, 10) Is there: Pr<sub>9</sub>?
10. (11, 13) Is there: Nu<sub>34</sub>?
11. (12, -) Change the code number into 39.
12. (13, -) Take: n, i'.
13. Turn to C. V.
14. (11, 15) Is there: Nu<sub>35</sub>?
15. Turn to Q<sub>27</sub>.

Values of the parameters in the changing programs:

<u>Input:</u>		<u>Output:</u>	
<u>Value</u>	<u>Sign:</u>	<u>Sign:</u>	<u>Value:</u>
$Pr_3 + Pr_6 + Pr_7$	= $Pr_k$	$Pr_k,$	$\Rightarrow Pr_3$
$Pr_1 + Pr_9$	= $Pr_m$	$Pr_m,$	$\Rightarrow Pr_1$
$A_{26}$	= $A_k$	$A_k,$	$\Rightarrow A_2$
$A_{26}$	= $A_m$	$A_m,$	$\Rightarrow A_3$
$Nu_{26}$	= $Nu_k$	$Nu_k,$	$\Rightarrow Nu_2$
$Nu_{26}$	= $Nu_m$	$Nu_m,$	$\Rightarrow Nu_3$
$P_{26}$	= $P_k$	$P_k,$	$\Rightarrow P_2$
$P_{26}$	= $P_m$	$P_m,$	$\Rightarrow P_3$
$A_{31}$	= $A_n$	$A_i,$	$\Rightarrow A_{19}$
$P_{31}$	= $P_n$	$P_i,$	$\Rightarrow P_{19}$
$P_{32}$	= $P_u$		
$N_{27}$	= $N_k$	$N_k,$	$\Rightarrow N_2$
$N_{27}$	= $N_i$	$N_i,$	$\Rightarrow N_{19}$

D. IV, (N<sub>28</sub>; fig. 4.; form. 10-11.)

1. (2, <4) Is there:  $Pr_2$  in the word group?
2. (3, <-) Take:  $k, k'$ .
3. < Turn to C. I.
4. (5, <7) Is there:  $Pr_5$ ?
5. (6, <-) Take:  $k, m'$ .

6. Turn to C. I.
7. (5, 8) Is there:  $Pr_7$ ?
8. (9, 15) Is there:  $Pr_8$ ?
9. (10, 12) Is there: A ?
10. (11, -) Take: k, k'.
11. Turn to C. II.
12. (10, 13) Is there: Nu?
13. (10, 14) Is there: P?
14. Turn to  $Q_{28}$
15. (16, -) Take: k, k'.
16. Turn to C. III.

Values of the parameters in the changing programs:

<u>Input:</u>		<u>Output:</u>	
<u>Value:</u>	<u>Sign:</u>	<u>Sign:</u>	<u>Value:</u>
$Pr_2 + Pr_5 + Pr_7$	= $Pr_k$	$Pr_k,$	$\Rightarrow Pr_2$
$Pr_8$	= $Pr_i$	$Pr_m,$	$\Rightarrow Pr_5$
$A_{29} + A_{35}$	= $A_k$	$A_k,$	$\Rightarrow A_{10}$
$Nu_{29}$	= $Nu_k$	$Nu_k,$	$\Rightarrow Nu_{10}$
$P_{29} + P_{35}$	= $P_k$	$P_k,$	$\Rightarrow P_{10}$
		$A_m,$	$\Rightarrow A_{12}$
		$Nu_m,$	$\Rightarrow Nu_{12}$
		$P_m,$	$\Rightarrow P_{12}$

$$N_{28} = N_k \Rightarrow N_{10}$$

$$N_{28} = N_i \Rightarrow N_{12}$$

D. V. (N<sub>30</sub>; fig. 5.; form. 12-13.)

1. (2, 3) Is there: Pr<sub>3</sub> in the word group?
2. Turn to C. I.
3. (2, 4) Is there: Pr<sub>6</sub>?
4. (2, 5) Is there: Pr<sub>7</sub>?
5. Turn to Q<sub>30</sub>.

Values of the parameters in the changing programs:

<u>Input:</u>		<u>Output:</u>	
<u>Value:</u>	<u>Sign:</u>	<u>Sign:</u>	<u>Value:</u>
Pr <sub>3</sub> +Pr <sub>6</sub> +Pr <sub>7</sub>	= Pr <sub>i</sub>	Pr <sub>i</sub> ,	=> Pr <sub>3</sub>
A <sub>30</sub>	= A <sub>i</sub>	A <sub>i</sub> ,	=> A <sub>13</sub>
Nu <sub>30</sub>	= Nu <sub>i</sub>	Nu <sub>i</sub> ,	=> Nu <sub>14</sub>
P <sub>30</sub>	= P <sub>i</sub>	P <sub>i</sub> ,	=> P <sub>14</sub>
N <sub>30</sub>	= N <sub>i</sub>	N <sub>i</sub> ,	=> N <sub>14</sub>

D. VI. (N<sub>31</sub>; fig. 6.; form. 14-17.)

1. (2, 4) Is there: Pr<sub>3</sub> in the word group?
2. (3, -) Take: k, k'.
3. Turn to C. I.
4. (2, 5) Is there: Pr<sub>6</sub>?

5. (6, 8) < Is there:  $Pr_5$ ?
6. (7, -) Take:  $m, m'$ .
7. Turn to C. I.
8. (6, 9) Is there:  $Pr_8$ ?
9. (10, 16) Is there:  $Pr_7$ ?
10. (11, 13) Is there:  $A$ ?
11. (12, -) Take:  $k, m, k', m'$ .
12. Turn to C. II.
13. (11, 14) Is there:  $Nu$ ?
14. (11, 15) Is there:  $P$ ?
15. Turn to  $Q_{42}$ .
16. (19, 17) Is there:  $A$ ?
17. (19, 18) Is there:  $Nu$ ?
18. (19, 21) Is there:  $P$ ?
19. (20, -) Take:  $k, n'$ .
20. < Turn to C. III.
21. Turn to  $Q_{31}$ .

Values of the parameters in the changing programs:

<u>Input:</u>		<u>Output:</u>	
<u>Value:</u>	<u>Sign:</u>	<u>Sign:</u>	<u>Value:</u>
$Pr_3 + Pr_6$	= $Pr_k$	$Pr_k,$	$\Rightarrow Pr_3$
$Pr_5 + Pr_8$	= $Pr_m$	$Pr_m,$	$\Rightarrow Pr_5$
$Pr_7$	= $Pr_i$		

$A_{30}$	=	$A_k$	$A_k,$	$\Rightarrow$	$A_{14}$
$Nu_{30}$	=	$Nu_k$	$Nu_k,$	$\Rightarrow$	$Nu_{14}$
$P_{30}$	=	$P_k$	$P_k,$	$\Rightarrow$	$P_{14}$
$A_{28}$	=	$A_m$	$A_m,$	$\Rightarrow$	$A_{18}$
$Nu_{28}$	=	$Nu_m$	$Nu_m,$	$\Rightarrow$	$Nu_{18}$
$P_{28}$	=	$P_m$	$P_m,$	$\Rightarrow$	$P_{18}$
			$A_n,$	$\Rightarrow$	$A_{30}$
			$Nu_n,$	$\Rightarrow$	$Nu_{30}$
			$P_n,$	$\Rightarrow$	$P_{30}$
$N_{31}$	=	$N_m$	$N_m,$	$\Rightarrow$	$N_{18}$
$N_{31}$	=	$N_k$	$N_k,$	$\Rightarrow$	$N_{14}$
			$N_n,$	$\Rightarrow$	$N_{30}$
			$N_p,$	$\Rightarrow$	$N_{42}$

D. VII. ( $N_{32}$ ; fig. 7.; form. 18-19.)

1. (2, 3) Is there:  $Pr_3$  in the word group?
2. Turn to C. I.
3. (2, 4) Is there:  $Pr_6$ ?
4. (2, 5) Is there:  $Pr_7$ ?
5. Turn to  $Q_{32}$ .

Values of the parameters in the changing programs:

<u>Input:</u>		<u>Output:</u>	
<u>Value:</u>	<u>Sign:</u>	<u>Sign:</u>	<u>Value:</u>
$Pr_3+Pr_6+Pr_7$	= $Pr_i$	$Pr_i,$	$\Rightarrow Pr_3$
$A_{32}$	= $A_i$	$A_i,$	$\Rightarrow A_{20}$
$Nu_{32}$	= $Nu_i$	$Nu_i,$	$\Rightarrow Nu_{20}$
$P_{32}$	= $P_i$	$P_i,$	$\Rightarrow P_{20}$
$N_{32}$	= $N_i$	$N_i,$	$\Rightarrow N_{20}$

D. VIII. ( $N_{33}$ ; fig. 8.; form. 20-25.)

1. (2, 4) Is there:  $Pr_3$  in the word group?
2. (3, -) Take: k, z, k', r'.
3. Turn to C. I.
4. (2, 5) Is there:  $Pr_6$ ?
5. (2, 6) Is there:  $Pr_7$ ?
6. (7, 9) Is there:  $Pr_1$ ?
7. (8, -) Take: k, v, m', s'.
8. Turn to C. I.
9. (7,10) Is there:  $Pr_9$ ?
10. (11,14) Is there:  $Nu_{48}$ ?
11. (12, -) Change its code number for 48.
12. (13, -) Take: k, m'.
13. Turn to C. V.



14. (15,16) Is there:  $Nu_{35}$ ?
15. Turn to C. VI.
16. Turn to Q<sub>33</sub>.

Values of the parameters in the changing programs:

<u>Value:</u>	<u>Input:</u>	<u>Sign:</u>	<u>Sign:</u>	<u>Output:</u>	<u>Value:</u>
$Pr_3+Pr_6+Pr_7$	=	$Pr_k$		$Pr_k,$	$\Rightarrow Pr_3$
$A_{31}$	=	$A_k$		$A_k,$	$\Rightarrow A_{20}$
$P_{31}$	=	$P_k$		$P_k,$	$\Rightarrow P_{20}$
$N_{33}$	=	$N_k$		$N_k,$	$\Rightarrow N_{20}$
				$A_m,$	$\Rightarrow A_{21}$
				$Nu_m,$	$\Rightarrow Nu_{21}$
$P_{32}$	=	$P_m$		$P_m,$	$\Rightarrow P_{21}$
				$N_m,$	$\Rightarrow N_{21}$
				$A_n,$	$\Rightarrow A_{32}$
				$P_n,$	$\Rightarrow P_{32}$
				$N_n,$	$\Rightarrow N_{32}$
				$A_p,$	$\Rightarrow A_{19}$
				$P_p,$	$\Rightarrow P_{19}$
				$N_p,$	$\Rightarrow N_{19}$
$Nu_{35}+Nu_{36}+Nu_{51}$	=	$Nu_z$		$Nu_r,$	$\Rightarrow Nu_{42}$
$Nu_{36}+Nu_{48}+Nu_{51}$	=	$Nu_v$		$Nu_i,$	$\Rightarrow Nu_{46}$
				$Nu_p,$	$\Rightarrow Nu_{49}$
				$Nu_s,$	$\Rightarrow Nu_{46}$

D. IX. (N<sub>34</sub>; fig. 9.; form. 26-28.)

1. (2, 8) Is there: Pr<sub>3</sub> in the word group?
2. (3, 5) Is there: Nu<sub>34</sub>?
3. (4, -) Change its code number for 42.
4. Turn to C. IV.
5. (3, 6) Is there: Nu<sub>35</sub>?
6. (7, -) Take: m, n'.
7. Turn to C. I.
8. (2, 9) Is there: Pr<sub>6</sub>?
9. (2,10) Is there: Pr<sub>7</sub>?
10. (11, 13) Is there: Pr<sub>1</sub>?
11. (12, - ) Take: k, k'.
12. Turn to C. I.
13. (11, 14) Is there: Pr<sub>9</sub>?
14. (15, 18) Is there: Nu<sub>34</sub>?
15. (16, - ) Change its code number for 35.
16. (17, - ) Take: n, m'.
17. Turn to C. V.
18. (15, 19) Is there: Nu<sub>35</sub>?
19. Turn to C. III.

Values of the parameters in the changing programs:

<u>Input:</u>			<u>Output:</u>		
<u>Value:</u>		<u>Sign:</u>	<u>Sign:</u>		<u>Value:</u>
$Pr_1+Pr_9$	=	$Pr_k$	$Pr_k,$	$\Rightarrow$	$Pr_1$
$Pr_3+Pr_6+Pr_7$	=	$Pr_m$	$Pr_n,$	$\Rightarrow$	$Pr_3$
$A_{26}$	=	$A_k$	$A_k,$	$\Rightarrow$	$A_{15}$
$Nu_{26}$	=	$Nu_k$	$Nu_k,$	$\Rightarrow$	$Nu_{15}$
$P_{26}$	=	$P_k$	$P_k,$	$\Rightarrow$	$P_{15}$
$A_{32}$	=	$A_m$	$A_m,$	$\Rightarrow$	$A_{32}$
$Nu_{32}$	=	$Nu_m$	$Nu_m,$	$\Rightarrow$	$Nu_{32}$
$P_{32}$	=	$P_m$	$P_m,$	$\Rightarrow$	$P_{32}$
$N_{34}$	=	$N_m$	$N_m,$	$\Rightarrow$	$N_{32}$
$A_{31}$	=	$A_n$	$A_n,$	$\Rightarrow$	$A_{20}$
$Nu_{31}$	=	$Nu_n$	$Nu_n,$	$\Rightarrow$	$Nu_{20}$
$P_{31}$	=	$P_n$	$P_n,$	$\Rightarrow$	$P_{20}$
$P_{32}$	=	$P_u$	$N_n,$	$\Rightarrow$	$N_{20}$

D. X. (N<sub>35</sub>; fig. 10.; form. 29-31.)

1. (2, 4) Is there:  $Pr_5$  in the word group?
2. (3, -) Take:  $i, i'$ .
3. Turn to C. I.
4. (2, 5) Is there:  $Pr_7$ ?
5. (2, 6) Is there:  $Pr_8$ ?

6. (7, 9) Is there:  $Pr_3$ ?
7. (8, -) Take:  $m, m'$ .
8. Turn to C. I.
9. (7, 10) Is there:  $Pr_6$ ?
10. Turn to C. III.

Values of the parameters in the changing programs:

<u>Input:</u>		<u>Output:</u>		
<u>Value:</u>		<u>Sign:</u>		<u>Value:</u>
$Pr_5+Pr_7+Pr_8$	=	$Pr_i$	$Pr_i,$	$\Rightarrow Pr_5$
$Pr_3+Pr_6$	=	$Pr_m$	$Pr_m,$	$\Rightarrow Pr_3$
$A_{28}$	=	$A_i$	$A_i,$	$\Rightarrow A_6$
$Nu_{28}$	=	$Nu_i$	$Nu_i,$	$\Rightarrow Nu_6$
$P_{28}$	=	$P_i$	$P_i,$	$\Rightarrow P_6$
$N_{35}$	=	$N_k$	$N_k,$	$\Rightarrow N_{32}$
$N_{35}$	=	$N_i$	$N_i,$	$\Rightarrow N_6$
$N_{35}$	=	$N_m$	$N_m,$	$\Rightarrow N_{20}$
$A_{32}$	=	$A_k$	$A_k,$	$\Rightarrow A_{32}$
$A_{32}$	=	$A_m$	$A_m,$	$\Rightarrow A_{20}$
$Nu_{32}$	=	$Nu_k$	$Nu_k,$	$\Rightarrow Nu_{32}$
$Nu_{32}$	=	$Nu_m$	$Nu_m,$	$\Rightarrow Nu_{20}$
$P_{32}$	=	$P_k$	$P_k,$	$\Rightarrow P_{32}$
$P_{32}$	=	$P_m$	$P_m,$	$\Rightarrow P_{20}$

D. XI. (N<sub>36</sub>; fig. 11.; form. 32-34.)

1. (2, 4) Is there: Pr<sub>1</sub> in the word group?
2. (3, -) Take: k, k'.
3. Turn to C. I.
4. (2, 5) Is there: Pr<sub>9</sub>?
5. (6,12) Is there: Pr<sub>3</sub>?
6. (7, 9) Is there: Nu<sub>35</sub>?
7. (8, -) Change its code number for 45.
8. Turn to C. IV.
9. (7,10) Is there: Nu<sub>43</sub>?
10. (11.-) Take: m, n'.
11. Turn to C. I.
12. (6,13) Is there: Pr<sub>6</sub>?
13. (6,14) Is there: Pr<sub>7</sub>?
14. (15,18) Is there: Nu<sub>35</sub>?
15. (16, -) Change its code number for 35.
16. (17, -) Take: n, m'.
17. Turn to C. V.
18. (15,19) Is there: Nu<sub>43</sub>?
19. Turn to C. III.

Values of the parameters in the changing programs:

<u>Input:</u>		<u>Output:</u>		
<u>Value:</u>		<u>Sign:</u>	<u>Sign:</u>	<u>Value:</u>
$Pr_1 + Pr_9$	=	$Pr_k$	$Pr_k,$	$\Rightarrow Pr_1$
$A_{29}$	=	$A_k$	$A_k,$	$\Rightarrow A_9$
$Nu_{29}$	=	$Nu_k$	$Nu_k,$	$\Rightarrow Nu_9$
$P_{29}$	=	$P_k$	$P_k,$	$\Rightarrow P_9$
$N_{36}$	=	$N_k$	$N_k,$	$\Rightarrow N_9$
$N_{36}$	=	$N_m$	$N_m,$	$\Rightarrow N_{32}$
$N_{36}$	=	$N_n$	$N_n,$	$\Rightarrow N_{20}$
$A_{32}$	=	$A_m$	$A_m,$	$\Rightarrow A_{32}$
$Nu_{32}$	=	$Nu_m$	$Nu_m,$	$\Rightarrow Nu_{32}$
$P_{32}$	=	$P_m$	$P_m,$	$\Rightarrow P_{32}$
$P_{32}$	=	$P_u$		
$A_{31}$	=	$A_n$	$A_n,$	$\Rightarrow A_{20}$
$Nu_{31}$	=	$Nu_n$	$Nu_n,$	$\Rightarrow Nu_{20}$
$P_{31}$	=	$P_n$	$P_n,$	$\Rightarrow P_{20}$

D. XII. ( $N_{37}$ ; fig. 12.; form. 35-42)

1. (2, 4) Is there:  $Pr_1$  in the word group?
2. (3, -) Take:  $i, i'$ .
3. Turn to C. I.
4. (2, 5) Is there:  $Pr_9$ ?

5. (6, 8) Is there:  $Pr_2$ ?
6. (7, -) Take:  $k, p'$ .
7. Turn to C. I.
8. (9, 11) Is there:  $Pr_5$ ?
9. (10, -) Take:  $k, s'$ ,
10. Turn to C. I.
11. (12, 18) Is there:  $Pr_3$ ?
12. (13, 15) Is there:  $Nu_{35}$ ?
13. (14, -) Change its code number for 45.
14. Turn to C. IV.
15. (13, 16) Is there:  $Nu_{43}$ ?
16. (17, -) Take:  $m, n'$ .
17. Turn to C. I.
18. (12, 19) Is there:  $Pr_6$ ?
19. (20, 31) Is there:  $Pr_7$ ?
20. (21, 23) Is there:  $Nu_{35}$ ?
21. (22, -) Change its code number for 45.
22. Turn to C. IV.
23. (21, 24) Is there:  $Nu_{43}$ ?
24. (25, 27) Is there: A?
25. (26, -) Take:  $k' = s'; m' = n'$ .
26. Turn to C. II.
27. (25, 28) Is there: Nu?
28. (25, 29) Is there: P?

29. (30, -) Change the code number of the noun for 43.
30. Turn to Q<sub>43</sub>.
31. (32,34) Is there: Pr<sub>8</sub>?
32. (33, -) Take: k, r'.
33. Turn to C. I.
34. (35,38) Is there: Nu<sub>35</sub>?
35. (36, -) Change its code number for 35.
36. (37, -) Take: n, m'.
37. Turn to C. V.
38. (35,39) Is there: Nu<sub>43</sub>?
39. Turn to C. III.

Values of the parameters in the changing programs:

<u>Value:</u>	<u>Input:</u>	<u>Sign:</u>	<u>Sign:</u>	<u>Output:</u>	<u>Value:</u>
Pr <sub>3</sub> +Pr <sub>6</sub>	=	Pr <sub>m</sub>	Pr <sub>m</sub> ,	⇒	Pr <sub>3</sub>
Pr <sub>1</sub> +Pr <sub>9</sub>	=	Pr <sub>i</sub>	Pr <sub>i</sub> ,	⇒	Pr <sub>1</sub>
Pr <sub>2</sub>	=	Pr <sub>p</sub>	Pr <sub>p</sub> ,	⇒	Pr <sub>2</sub>
Pr <sub>8</sub>	=	Pr <sub>r</sub>	Pr <sub>r</sub> ,	⇒	Pr <sub>8</sub>
Pr <sub>5</sub>	=	Pr <sub>s</sub>	Pr <sub>s</sub> ,	⇒	Pr <sub>5</sub>
Pr <sub>7</sub>	=	Pr <sub>t</sub>	Pr <sub>t</sub> ,	⇒	Pr <sub>7</sub>
			Pr <sub>n</sub> ,	⇒	Pr <sub>3</sub>
A <sub>29</sub> +A <sub>35</sub>	=	A <sub>k</sub>	A <sub>k</sub> ,	⇒	A <sub>46</sub>
A <sub>29</sub> +A <sub>35</sub>	=	A <sub>i</sub>	A <sub>i</sub> ,	⇒	A <sub>9</sub>



Nu <sub>29</sub>	=	< Nu <sub>k</sub>	Nu <sub>k</sub> ,	=>	Nu <sub>46</sub>
Nu <sub>29</sub>	=	< Nu <sub>i</sub>	Nu <sub>i</sub> ,	=>	Nu <sub>9</sub>
P <sub>29</sub>	=	< P <sub>k</sub>	P <sub>k</sub> ,	=>	P <sub>46</sub>
P <sub>29</sub>	=	< P <sub>i</sub>	P <sub>i</sub> ,	=>	P <sub>9</sub>
N <sub>37</sub>	=	N <sub>k</sub>	N <sub>k</sub> ,	=>	N <sub>29</sub>
N <sub>37</sub>	=	N <sub>i</sub>	N <sub>i</sub> ,	=>	N <sub>9</sub>
N <sub>37</sub>	=	N <sub>m</sub>	N <sub>m</sub> ,	=>	N <sub>32</sub>
N <sub>37</sub>	=	N <sub>n</sub>	N <sub>n</sub> ,	=>	N <sub>20</sub>
			N <sub>p</sub> ,	=>	N <sub>10</sub>
			N <sub>r</sub> ,	=>	N <sub>28</sub>
			N <sub>s</sub> ,	=>	N <sub>12</sub>
A <sub>32</sub>	=	A <sub>m</sub>	A <sub>m</sub> ,	=>	A <sub>32</sub>
Nu <sub>32</sub>	=	Nu <sub>m</sub>	Nu <sub>m</sub> ,	=>	Nu <sub>32</sub>
P <sub>32</sub>	=	P <sub>m</sub>	P <sub>m</sub> ,	=>	P <sub>32</sub>
P <sub>32</sub>	=	P <sub>u</sub>			
A <sub>31</sub>	=	A <sub>n</sub>	A <sub>n</sub> ,	=>	A <sub>20</sub>
Nu <sub>31</sub>	=	Nu <sub>n</sub>	Nu <sub>n</sub> ,	=>	Nu <sub>20</sub>
P <sub>31</sub>	=	P <sub>n</sub>	P <sub>n</sub> ,	=>	P <sub>20</sub>
			A <sub>p</sub> ,	=>	A <sub>10</sub>
			Nu <sub>p</sub> ,	=>	Nu <sub>10</sub>
			P <sub>p</sub> ,	=>	P <sub>10</sub>
			A <sub>r</sub> ,	=>	A <sub>37</sub>
			Nu <sub>r</sub> ,	=>	Nu <sub>37</sub>

$P_r,$	$\Rightarrow$	$P_{37}$
$A_s,$	$\Rightarrow$	$A_{12}$
$Nu_s$	$\Rightarrow$	$Nu_{12}$
$P_s$	$\Rightarrow$	$P_{12}$

D. XIII. (N<sub>38</sub>; fig. 13.; form. 43-46)

1. (2, 4) Is there: Pr<sub>3</sub>?
2. (3, -) Take: n, n'.
3. Turn to. C. I.
4. (2, 5) Is there: Pr<sub>6</sub>?
5. (2, 6) Is there: Pr<sub>7</sub>?
6. (7, 9) Is there: Pr<sub>1</sub>?
7. (8, -) Take: p, p'.
8. Turn to C. I.
9. (7,10) Is there: Pr<sub>9</sub>?
10. (11,13) Is there: Nu<sub>34</sub>?
11. (12, -) Change its code number for 39.
12. Turn to C. V.
13. (11,14) Is there: Nu<sub>35</sub>?
14. Turn to C, III.

Values of the parameters in the changing programs:

<u>Input:</u>		<u>Output:</u>	
<u>Value:</u>	<u>Sign:</u>	<u>Sign:</u>	<u>Value:</u>
$Pr_3 + Pr_6 + Pr_7$	=	$Pr_n$	$\Rightarrow Pr_3$
$Pr_1 + Pr_9$	=	$Pr_p$	$\Rightarrow Pr_1$
$A_{32}$	=	$A_k$	$\Rightarrow A_{19}$
$Nu_{32}$	=	$Nu_k$	$\Rightarrow Nu_{19}$
$P_{32}$	=	$P_k$	$\Rightarrow P_{19}$
$P_{32}$	=	$P_u$	
$N_{38}$	=	$N_k$	$\Rightarrow N_{19}$
$N_{38}$	=	$N_m$	$\Rightarrow N_{27}$
$N_{38}$	=	$N_i$	$\Rightarrow N_{19}$
$N_{38}$	=	$N_n$	$\Rightarrow N_2$
$N_{38}$	=	$N_p$	$\Rightarrow N_3$
$A_{26}$	=	$A_m$	$\Rightarrow A_{26}$
$A_{26}$	=	$A_n$	$\Rightarrow A_2$
$A_{26}$	=	$A_p$	$\Rightarrow A_3$
$Nu_{26}$	=	$Nu_m$	$\Rightarrow Nu_{26}$
$Nu_{26}$	=	$Nu_p$	$\Rightarrow Nu_3$
$Nu_{26}$	=	$Nu_n$	$\Rightarrow Nu_2$
$P_{26}$	=	$P_m$	$\Rightarrow P_{26}$
$P_{26}$	=	$P_n$	$\Rightarrow P_2$
$P_{26}$	=	$P_p$	$\Rightarrow P_3$

$$A_{31} = A_i \quad A_i, \Rightarrow A_{19}$$

$$P_{31} = P_i \quad P_i, \Rightarrow P_{19}$$

D. XIV. (N<sub>39</sub>; fig. 14.; form. 47-49)

1. (2, 4) Is there: Pr<sub>1</sub>?
2. (3, -) Take: k, k'.
3. Turn to C. I.
4. (2, 5) Is there: Pr<sub>9</sub>?
5. (6,12) Is there: Pr<sub>3</sub>?
6. (7, 9) Is there: Nu<sub>34</sub>?
7. (8, -) Change its code number for 41.
8. Turn to C. IV.
9. (7,10) Is there: Nu<sub>35</sub>?
10. (11,-) Take: m, n'.
11. Turn to C. I.
12. (6,13) Is there: Pr<sub>6</sub>?
13. (6,14) Is there: Pr<sub>7</sub>?
14. (15,17) Is there: Nu<sub>34</sub>?
15. (16, -) Change its code number for 35.
16. Turn to C. V.
17. (15,18) Is there: Nu<sub>35</sub>?
18. Turn to C. III.

Values of the parameters in the changing programs:

<u>Input:</u>			<u>Output:</u>		
<u>Value:</u>		<u>Sign:</u>	<u>Sign:</u>		<u>Value:</u>
$Pr_1 + Pr_9$	=	$Pr_k$	$Pr_k,$	$\Rightarrow$	$Pr_1$
$Pr_3 + Pr_6 + Pr_7$	=	$Pr_m$	$Pr_m,$	$\Rightarrow$	$Pr_3$
			$Pr_n,$	$\Rightarrow$	$Pr_3$
$A_{26}$	=	$A_k$	$A_k,$	$\Rightarrow$	$A_3$
$Nu_{26}$	=	$Nu_k$	$Nu_k,$	$\Rightarrow$	$Nu_3$
$P_{26}$	=	$P_k$	$P_k,$	$\Rightarrow$	$P_3$
$N_{39}$	=	$N_k$	$N_k,$	$\Rightarrow$	$N_3$
$N_{39}$	=	$N_m$	$N_m,$	$\Rightarrow$	$N_{32}$
$N_{39}$	=	$N_i$	$N_i,$	$\Rightarrow$	$N_{32}$
$N_{39}$	=	$N_n$	$N_n,$	$\Rightarrow$	$N_{20}$
$A_{32}$	=	$A_m$	$A_m,$	$\Rightarrow$	$A_{32}$
$Nu_{32}$	=	$Nu_m$	$Nu_m,$	$\Rightarrow$	$Nu_{32}$
$Nu_{32}$	=	$Nu_n$	$Nu_n,$	$\Rightarrow$	$Nu_{20}$
$P_{32}$	=	$P_m$	$P_m,$	$\Rightarrow$	$P_{32}$
$P_{32}$	=	$P_u$			
$A_{31}$	=	$A_n$	$A_n,$	$\Rightarrow$	$A_{20}$
$P_{31}$	=	$P_n$	$P_n,$	$\Rightarrow$	$P_{20}$

D. XV. (N<sub>40</sub>; fig. 15.; form. 50-51.)

1. (2, 4) Is there: Pr<sub>3</sub>?
2. (3, -) Take: m, m'.
3. Turn to C. I.
4. (2, 5) Is there: Pr<sub>6</sub>?
5. (2, 6) Is there: Pr<sub>7</sub>?
6. Turn to C. III.

Values of the parameters in the changing programs:

<u>Input:</u>		<u>Output:</u>	
<u>Value:</u>	<u>Sign:</u>	<u>Sign:</u>	<u>Value:</u>
Pr <sub>3</sub> +Pr <sub>6</sub> +Pr <sub>7</sub>	= Pr <sub>m</sub>	Pr <sub>m</sub> ,	⇒ Pr <sub>3</sub>
A <sub>7</sub>	= A <sub>k</sub>	A <sub>k</sub> ,	⇒ A <sub>7</sub>
Nu <sub>7</sub>	= Nu <sub>k</sub>	Nu <sub>k</sub> ,	⇒ Nu <sub>7</sub>
P <sub>7</sub>	= P <sub>k</sub>	P <sub>k</sub> ,	⇒ P <sub>7</sub>
N <sub>40</sub>	= N <sub>k</sub>	N <sub>k</sub> ,	⇒ N <sub>7</sub>
N <sub>40</sub>	= N <sub>m</sub>	N <sub>m</sub> ,	⇒ N <sub>8</sub>
A <sub>8</sub>	= A <sub>m</sub>	A <sub>m</sub> ,	⇒ A <sub>8</sub>
Nu <sub>8</sub>	= Nu <sub>m</sub>	Nu <sub>m</sub> ,	⇒ Nu <sub>8</sub>
P <sub>8</sub>	= P <sub>m</sub>	P <sub>m</sub> ,	⇒ P <sub>8</sub>

D. XVI. (N<sub>41</sub>; fig. 16.; form. 52-53.)

1. (2, 4) Is there: Pr<sub>1</sub>?
2. (3, -) Take: k, k'.
3. Turn to C. I.
4. (2, 5) Is there: Pr<sub>9</sub>?
5. (6, 8) Is there: Nu<sub>35</sub>?
6. (7, -) Change its code number for 44.
7. Turn to C. V.
8. (6, 9) Is there: Nu<sub>43</sub>?
9. Turn to C. III.

Values of the parameters in the changing programs:

<u>Value:</u>	<u>Input:</u>	<u>Sign:</u>	<u>Output:</u>	<u>Value:</u>
Pr <sub>1</sub> +Pr <sub>9</sub>	=	Pr <sub>k</sub>	Pr <sub>k</sub> ,	⇒ Pr <sub>1</sub>
A <sub>29</sub> +A <sub>35</sub>	=	A <sub>k</sub>	A <sub>k</sub> ,	⇒ A <sub>9</sub>
Nu <sub>29</sub>	=	Nu <sub>k</sub>	Nu <sub>k</sub> ,	⇒ Nu <sub>9</sub>
P <sub>29</sub> +P <sub>35</sub>	=	P <sub>k</sub>	P <sub>k</sub> ,	⇒ P <sub>9</sub>
N <sub>41</sub>	=	N <sub>k</sub>	N <sub>k</sub> ,	⇒ N <sub>9</sub>
N <sub>41</sub>	=	N <sub>m</sub>	N <sub>m</sub> ,	⇒ N <sub>19</sub>
N <sub>41</sub>	=	N <sub>i</sub>	N <sub>i</sub> ,	⇒ N <sub>19</sub>
A <sub>31</sub>	=	A <sub>i</sub>	A <sub>i</sub> ,	⇒ A <sub>19</sub>
P <sub>31</sub>	=	P <sub>i</sub>	P <sub>i</sub> ,	⇒ P <sub>19</sub>
A <sub>32</sub>	=	A <sub>m</sub>	A <sub>m</sub> ,	⇒ A <sub>19</sub>

$$\begin{array}{rclcl}
 \text{Nu}_{32} & = & \text{Nu}_m & \text{Nu}_m, & \Rightarrow & \text{Nu}_{19} \\
 \text{P}_{32} & = & \text{P}_m & \text{P}_m, & \Rightarrow & \text{P}_{19} \\
 \text{P}_{32} & = & \text{P}_u & & & 
 \end{array}$$

As it is to be seen from the sketches of programs listed above, there are several instructions common in each of them (e.g.: ones for finding out the presence or absence of some substantives or prepositions with the same code number). The program for the computer can be made shorter by starting with a common algorithm for these common operations and branching out the program on the point where the analysis necessitates various operations for each of the nominal forms ("switch").

#### b) Changing Programs

The changing programs contain the following instructions:<sup>1/</sup>

##### C. I.

1. (2, -)  $\text{Pr}_i = \text{Pr}_i$ ,<sup>2/</sup>
2. (3, 4) Is there: A in the word group?
3. (4, -)  $\text{A}_i = \text{A}_i$ ,
4. (5, 6) Is there:  $\text{Nu}_i$ ?
5. (6, -)  $\text{Nu}_i = \text{Nu}_i$ ,.

<sup>1/</sup> In the changing programs the signs of the parameters given at the end of the various determining programs are to be found. Replacing the signs with the corresponding values the values of the parameters can be got; the values of parameters corresponding to the signs marked by "'" give the changed code number of the morpheme.

<sup>2/</sup> i.e.: change the value corresponding to the sign  $\text{Pr}_i$  for the value corresponding to the sign  $\text{Pr}_i$ ,.



6. (7, 8) Is there:  $Nu_z$ ?
7. (8, -)  $Nu_z = Nu_r, .$
8. (9, 10) Is there:  $Nu_v$ ?
9. (10, -)  $Nu_v = Nu_s, .$
10. (11, 12) Is there: P?
11. (12, -)  $P_i = P_i, .$
12. (13, -)  $N_i = N_i, .$
13. (14, -) Is the changed code number of the noun bigger than 24?
14. Turn to the program "Q" corresponding to this code number.

### C. II.

1. (2, 3) Is there:  $A_k$  in the word group?
2. (3, -)  $A_k = A_k, .$
3. (4, 5) Is there:  $Nu_k$ ?
4. (5, -)  $Nu_k = Nu_k, .$
5. (6, 7) Is there:  $P_k$ ?
6. (7, -)  $P_k = P_k, .$
7. (8, 9) Is there:  $A_m$ ?
8. (9, -)  $A_m = A_m, .$
9. (10, 11) Is there:  $Nu_m$ ?
10. (11, -)  $Nu_m = Nu_m, .$
11. (12, 13) Is there:  $P_m$ ?
12. (13, -)  $P_m = P_m, .$

13. (14,16) Is there  $k'$  among the changed code numbers?
14. (15, -)  $N_i = N_k, .$
15.  $Pr_i = Pr_k, .$
16. (17, -)  $N_i = N_m, .$
17.  $Pr_i = Pr_m, .$

C. III.

1. (2, 3) Is there:  $A_k$  in the word group?
2. (3, -)  $A_k = A_k, .$
3. (4, 5) Is there:  $Nu_k?$
4. (5, -)  $Nu_k = Nu_k, .$
5. (6, 7) Is there:  $P_k?$
6. (7, -)  $P_k = P_k, .$
7. (8, 9) Is there:  $A_m?$
8. (9, -)  $A_m = A_m, .$
9. (10,11) Is there:  $Nu_m?$
10. (11, -)  $Nu_m = Nu_m, .$
11. (12,13) Is there:  $P_m?$
12. (13, -)  $P_m = P_m, .$
13. (14,15) Is there  $k'$  among the changed code numbers?
14. (17, -)  $N_k = N_k, .$
15. (16,19) Is there  $m'$  among the changed code numbers?
16. (17, -)  $N_m = N_m, .$

17. (18, -) Is the changed code number of the noun bigger than 24?
18. Turn to the program "Q" corresponding to this code number.
19. Turn to the program "Q" corresponding to the original code number of the noun.

C. IV.

1. (2, -) Change the code number of the preposition for 3.
2. (3, 4) Is there:  $A_n$ ?
3. (4, -)  $A_n = A_n, .$
4. (5, 6) Is there:  $P_m$ ?
5. (6, -)  $P_m = P_n, .$
6. (7, 8) Is there:  $P_n$ ?
7. (8, -)  $P_n = P_n, .$
8.  $N_m = N_n, .$

C. V.

1. (2, 3) Is there:  $A_i$  in the word group?
2. (3, -)  $A_i = A_i, .$
3. (4, 5) Is there:  $P_i$ ?
4. (5, -)  $P_i = P_i, .$
5. (6, 7) Is there:  $P_u$ ?
6. (7, -)  $P_u = P_i, .$
7. (8, -)  $N_i = N_i, .$

8. (9, -) Is the changed code number of the noun bigger than 24?
9. Turn to the program "Q" corresponding to this code number.

C. VI.

1. (2, 8) Is the first member of the word group:  $Nu_z$ ?
2. (3, 4) Is there:  $A_k$  ?
3. (4, -)  $A_k = A_n, .$
4. (5, 6) Is there:  $P_k$  ?
5. (6, -)  $P_k = P_n,$
6. (7, -)  $N_i = N_n, .$
7. Turn to the program "Q" corresponding to the changed code number of the noun.
8. (9,14) Is the first member of the word group:  $P_m$ ?
9. (10,-)  $P_m = P_p, .$
10. (11,-)  $Nu_z = Nu_p, .$
11. (12,13) Is there:  $A_k$  ?
12. (13, -)  $A_k = A_p, .$
13.  $N_i = N_p, .$
14. (15, -)  $Nu_z = Nu_r, .$
15. (16,17) Is there:  $A_k$  ?
16. (17, -)  $A_k = A_k, .$
17. (18,19) Is there:  $P_k$  ?
18. (19, -)  $P_k = P_k, .$
19.  $N_i = N_k, .$

## 6. The efficiency of the Programs

The method of analysis and programming reviewed above gives possibility of clear the role of the homonym substantive forms and the government of the prepositions in 87 % of the cases. Partial solution is received in 11 %, and the analysis does not give any solution in 1,5 % of the cases. It means that "Q" programs for further analysis are needed in 13 % of the cases. The cases where the role of the homonym noun form cannot be cleared because there is no preposition or other substantive in the given word group with it, are put among ones giving no solution. As it is to be seen from the programs listed above, the role of the given form cannot be cleared but with the help of "Q" programs in such cases too.

The proportions mentioned above have been calculated on the base of the number of the "steps" necessary to the solution without taking into account the frequency of occurrence of the single forms in the word groups, and so they must be regarded as theoretical ones. Considering the fact that the more typical a form is the more frequently it occurs in texts and that noun forms are relatively rare without any prepositions and other substantives, the practical results are likely to be more favourable than the ones mentioned above.

## 2. The Effectiveness of the Program

The extent of the program and its effectiveness in providing a possible alternative to the current system of care for the mentally ill has been evaluated. The program has been found to be effective in providing care for the mentally ill and in reducing the cost of care. The program has been found to be effective in providing care for the mentally ill and in reducing the cost of care. The program has been found to be effective in providing care for the mentally ill and in reducing the cost of care. The program has been found to be effective in providing care for the mentally ill and in reducing the cost of care.

The program has been found to be effective in providing care for the mentally ill and in reducing the cost of care. The program has been found to be effective in providing care for the mentally ill and in reducing the cost of care. The program has been found to be effective in providing care for the mentally ill and in reducing the cost of care. The program has been found to be effective in providing care for the mentally ill and in reducing the cost of care. The program has been found to be effective in providing care for the mentally ill and in reducing the cost of care.

ALGORITHMUS UND PROGRAMM FÜR DAS AUFSUCHEN EINES  
WORTES IM SPEICHER

F. Kiefer

Nach Y.Bar-Hillel (1) sei die automatische Übersetzung vor allem eine Frage der Speicherkapazität. Wenn wir heute diese Behauptung in einer anderen Beleuchtung sehen (sie stammt ja eigentlich aus dem Jahre 1954, d.h. aus einer Zeit, wo die Arbeiten an maschineller Übersetzung kaum begonnen haben), die Speicherkapazität blieb doch ein Problem, hauptsächlich in den kleineren Ländern, die über entsprechende Maschinen mit Riesenspeicher nicht verfügen.

Wollte man z.B. aus dem Englischen ins Ungarische übersetzen, so müsste man im Speicher ungefähr 500 000 Paradigmen und noch je 2-3 Ableitungen unterbringen, was die englische Sprache betrifft, und ungefähr 100 000 Paradigmen und je 10-15 Ableitungen, was die ungarische Sprache betrifft.

Wenn nur der Wortschatz einer Fachsprache herangezogen wird und dadurch der Umfang des Wörterbuches beträchtlich vermindert wird, so bleibt das Problem der Kapazität doch bestehen. Wollte man z.B. mathematische Fachtexte aus dem Russischen ins Ungarische übersetzen, so bestünde das "Wörterbuch" in der Maschine aus ungefähr 2500 russischen Paradigmen mit je 10-15 Ableitungen und den entsprechenden ungarischen Paradigmen mit ihren Ableitungen.

Es hat daher nicht nur einen theoretischen, sondern auch einen praktischen Wert, wenn man die Wörter im Speicher irgendwie "verdichtet" darstellen kann. Der Grundgedanke unseres Verfahrens stammt von W.S.Cooper (2), sein Algorithmus

wurde aber in mehreren Richtungen modifiziert und das entsprechende Programm auf der Rechenanlage M-3 (3) experimentell kontrolliert.

Nach der Eingabe sind die russischen Wörter (eigentlich nur Wortstämme und einige speziellen Wörter) in folgender Weise im Speicher untergebracht:

a) Ein Wort der Rechenanlage M-3 besteht aus 31 Dualstellen, die erste Stelle steht für das Vorzeichen. Ein Buchstabe beansprucht 6 bits, d.h. in einem Wort können 5 Buchstaben untergebracht werden. (Nach der Telex-Umcodierung eigentlich 6, das wurde aber mit Hilfe eines Unterprogramms verändert.)

b) Nach einer statistischen Angabe enthält das längste russische Wort 18 Buchstaben (4). Um ein russisches Wort im Speicher unterzubringen, braucht man daher höchstens vier Adressen. Gehören mehrere Adressen zu einem russischen Wort, so wird die Zusammengehörigkeit durch das Vorzeichen angegeben. Die erste Adresse bekommt nämlich positives Vorzeichen, die weiteren (die zweite, dritte, vierte Adresse) negative Vorzeichen.

c) Die Wörter sind nach ansteigender Grösse geordnet, und zwar in Gruppen nach der Länge (Länge = Zahl der Buchstaben) der Wörter, d.h. es gibt insgesamt 18 Gruppen, da das längste russische Wort - wie schon erwähnt - aus 18 Buchstaben besteht.

d) Vor der Eingabe wurde eine ausführliche Statistik über die einzugebenden Wörter verfertigt, einerseits um feststellen zu können, wieviel Wörter ein, zwei, drei, ..., achtzehn Buchstaben enthalten und andererseits auch über die "grammatische Häufigkeit" der vorhandenen Texte Auskunft zu bekommen. Auf diese Weise wird die unter c) genannte Einteilung nach Gruppen ermöglicht, andererseits bekommt man auch darüber Angaben, welche grammatische Regeln im Algorithmus in erster Linie zu berücksichtigen sind.



## Die Unterscheidungskomponenten

In jeder der genannten Gruppen sollen vorerst die sogenannten Unterscheidungskomponenten festgestellt werden. Unter Unterscheidungskomponente verstehen wir eine Zahl, die angibt, im wievielten Einer sich das  $i$ -te Wort vom  $(i-1)$ -ten Wort vom Wortanfang gerechnet unterscheidet, ist diese Zahl  $k$ , so gilt für die Unterscheidungskomponente

$$A = Z - (k-1)$$

wo  $Z$  die Anzahl der Dualstellen in einem Wort bezeichnet. Da bei der Rechenanlage M-3 ein Wort aus 30 Dualstellen besteht (das Vorzeichen lassen wir ausser acht), so gilt nun

$$A = 36 - (k-1)$$

wo wir die 30 im Achtersystem angeben.

Die Unterscheidungskomponente kann folgendermassen bestimmt werden:

Es seien

$$s_1, s_2, s_3, \dots, s_n$$

die Wörter nach ansteigender Grösse geordnet. Bezeichnen wir das Komplement dieser Wörter mit  $\bar{s}_i$  (wir verstehen hier unter Komplement die Zahl, welche sich aus der ursprünglichen Zahl ergibt, indem man ihr die Eins mit Null und die Null mit Eins in jeder Stelle ersetzt). Bezeichnen wir mit  $s_0$  das Wort, welches "leer" ist, d.h. aus lauter Nullen besteht. Bilden wir nun die folgenden Wörter

$$b_i = s_i \wedge \bar{s}_{i-1} \text{ für } 1 \leq i \leq n$$

Die logische Multiplikation führt zu den Zahlen  $b_i$ , in welchen der Stellenwert der ersten Eins zu bestimmen ist; wird dieser vom Wortende her bestimmt, so erhält man den erwünschten Wert von  $A$ .

Zur Bestimmung der ersten Eins in den Zahlen  $b_i$  scheinen zwei Methoden geeignet zu sein:

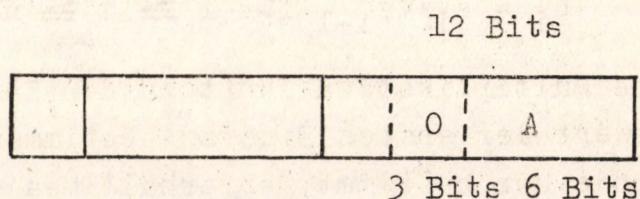
a) Man kann die Zahlen  $b_i$  als Mantissen von in Form einer Gleitkommazahl geschriebenen Zahlen betrachten und diese Zahlen normalisieren. Der auf diese Weise erhaltene Exponent wird eben die laufende Nummer des die erste Eins enthaltenden Stellenwertes sein.

b) Mit Hilfe eines Zyklus liesse sich die erste Eins auch feststellen, indem man nämlich jede Stelle (bit) des Wortes prüft, ob in ihr eine Eins steht oder nicht.

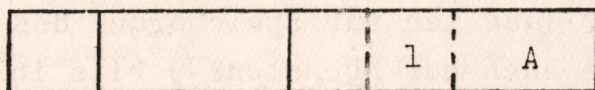
In beiden Fällen muss natürlich noch die Unterscheidungskomponente  $A$  berechnet werden, da hier nur der Wert von  $k$  bestimmt wird.

Da die meisten Wörter nicht in einer Adresse untergebracht werden können, muss man angeben, auf welche Adresse sich die Unterscheidungskomponente  $A$  bezieht. Die Unterscheidungskomponente  $A$ , deren maximaler Wert 36 beträgt (im Achtersystem), steht am Ende des Wortes und nimmt insgesamt höchstens 6 bits in Anspruch. Für die Unterbringung weiterer Informationen stehen uns noch in der zweiten Adresse des Wortes 6 bits zur Verfügung. Bezeichnen wir die Anzahl der Buchstaben mit  $n$ , so macht folgendes Schema die Darstellung der Unterscheidungskomponenten im Worte ersichtlich. Die vor der Unterscheidungskomponente stehende Zahl (0, 1, 2 oder 3) gibt an, ob sich die Unterscheidungskomponente im Falle von Wörtern mit mehr Adressen auf die erste, zweite, dritte oder vierte Adresse bezieht.

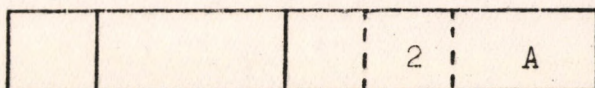
a) Bei Wörtern, bei denen sich die Unterscheidungskomponente auf die erste Adresse bezieht, gilt



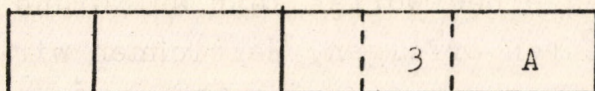
b) Bei Wörtern, bei denen sich die Unterscheidungskomponente auf die zweite Adresse bezieht, gilt



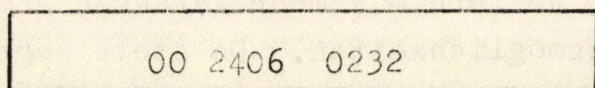
c) Bei Wörtern, bei denen sich die Unterscheidungskomponente auf die dritte Adresse bezieht, gilt



d) Bei Wörtern, bei denen sich die Unterscheidungskomponente auf die vierte Adresse bezieht, gilt



In den Wörtern im Speicher, in welche wir die Unterscheidungskomponenten untergebracht haben, steht in der ersten Adresse die Adresse des entsprechenden ungarischen Wortes (wortstammes). Ist z.B. die Unterscheidungskomponente eines russischen Wortes w 32, bezieht sich diese auf die dritte Adresse und ist die Adresse des entsprechenden ungarischen Wortes 2406, so gilt



Die Feststellung der Unterscheidungskomponenten ist auch im Falle eines Wortes mit vier Adressen im allgemeinen nicht langwierig. In meisten Fällen kann man sie schon in der ersten Adresse finden.

Nachdem wir die Unterscheidungskomponenten festgestellt haben, brauchen wir die russischen Wörter nicht speichern. Der grosse ökonomische Wert dieses Verfahrens ist schon daraus ersichtlich, dass man zur Speicherung des längsten russischen Wortes auch nur höchstens 9 bits in Anspruch nehmen zu braucht.

Dieses Verfahren ist aber für die Speicherung der ungarischen Wörter nicht geeignet, da man an Hand der Unterscheidungskomponenten die Wörter nicht mehr rekonstruieren vermag.

### Das Aufsuchen eines Wortes

Das Aufsuchen eines Wortes kann auf Grund der Unterscheidungskomponenten erfolgen. Bezeichnen wir das unbekannte Wort mit  $X$ . Vorerst muss festgestellt werden, zu welcher Gruppe das unbekannte Wort gehört. Bei der Eingabe wird die Zahl der Buchstaben eines Wortes stets gezählt. So wissen wir nun, ob das unbekannte Wort unter den Wörtern mit ein-, zwei, drei- usw. Buchstaben zu suchen ist.

Nachdem die Gruppe festgestellt ist, muss geprüft werden, ob in den Stellen, die durch die Unterscheidungskomponenten bestimmt werden, im unbekanntem Wort Einer stehen. Das kann durch logische Multiplikation erfolgen.

Im Falle von Wörtern mit mehr Adressen ergeben sich wieder Vereinfachungsmöglichkeiten. Da stets angegeben ist, auf welche Adresse sich die Unterscheidungskomponente bezieht, braucht nur eine Adresse untersucht zu werden.

Aus dem Ablaufbild des Algorithmus ist ersichtlich (S. Abbildung), dass wir die laufende Nummer des unbekanntes Wortes  $X$  erhalten. Das heisst, wenn das unbekannte Wort  $X$  aus  $r$  Buchstaben besteht und die Adresse des ersten Wortes in der Gruppe derjenigen Wörter, die aus  $r$  Buchstaben bestehen,

$\alpha_1$  ist, dann ergibt unser Algorithmus die laufende Nummer  $\eta$  des Wortes X. N bedeutet die Anzahl der Wörter in der entsprechenden Gruppe,  $A_i$  die entsprechenden Unterscheidungskomponenten. In der Abbildung bedeutet  $A_i$  die Unterscheidungskomponente, die Adresse  $\eta$  zählt die Wörter N ist die Anzahl der Wörter in der entsprechenden Gruppe, U und V sind "Arbeitszellen", n ist die Zahl der Buchstaben im gesuchten Wort.

Die Adresse des gesuchten Wortes ist im Falle von Wörtern mit einer Adresse (Buchstabenanzahl kleiner als 6)

$$\alpha_1 + (\eta - 1)$$

im allgemeinen aber

$$\alpha_1 + (\eta \cdot l - 1)$$

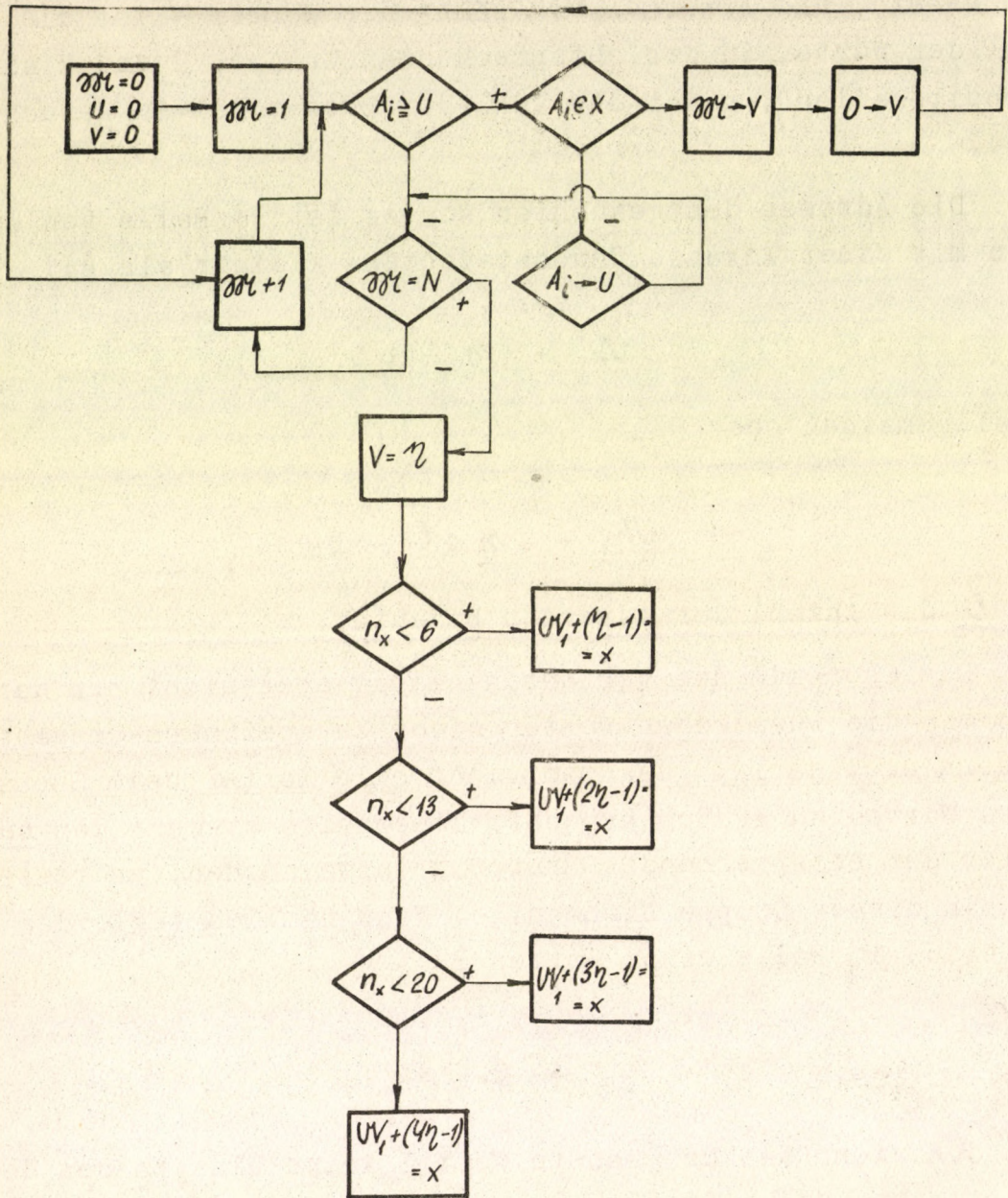
wo  $l$  die Anzahl der Adressen bedeutet.

Die Ökonomie des Verfahrens liegt aber nicht nur darin, dass man die russischen Wörter nicht zu speichern braucht, sondern auch darin, dass man nicht alle Wörter beim Suchen eines Wortes zu prüfen braucht. Einerseits braucht man nur die Wörter der entsprechenden Gruppe zu untersuchen, andererseits auch in dieser Gruppe nicht alle. Wenn nämlich (Vgl. das Ablaufbild) im Falle eines Wortes  $S_i$

$$A_i \notin X$$

ist, d.h. wenn das unbekannte Wort X keine Eins in der dem Wert  $A_i$  entsprechenden Stelle enthält, so braucht man mit dem nächstfolgenden Wort  $s_{i+1}$  nur dann den Vergleich zu vollführen, wenn die Unterscheidungskomponente dieses Wortes grösser ist als die Unterscheidungskomponente des Wortes  $s_i$ .

# Das Aufsuchen eines Wortes



Uns stehen zwar in dieser Hinsicht noch keine genaue statistischen Angaben zur Verfügung, aber im Falle von 20 "zufallsweise" generierten Wörtern brauchte man nur 6 Fälle zu untersuchen. Die Vermutung scheint berechtigt zu sein, dass man im allgemeinen nur ungefähr die Hälfte der Wörter zum Vergleich heranzuziehen braucht.

Beispiel:

Es seien im Speicher (nach ansteigender Grösse geordnet) die folgenden Wörter

$$s_1 = 00101101$$

$$s_2 = 00110010$$

$$s_3 = 01001101$$

$$s_4 = 01101010$$

$$s_5 = 01110100$$

$$s_6 = 01111000$$

$$s_6 = 10101000$$

$$s_8 = 11010101$$

Nehmen wir zu diesen noch  $s_0$

$$s_0 = 00000000,$$

das wird im Algorithmus die Anzahl der Wörter nicht beeinflussen.

Die Unterscheidungskomponente des Wortes  $s_1$  beträgt 6, denn

laufende Nummer	8	7	6	5	4	3	2	1
$s_0$	0	0	0	0	0	0	0	0
$s_1$	0	0	1	0	1	1	0	1

Im Worte  $s_1$  steht die erste Eins in der 6-ten Stelle vom Wortende gerechnet, in welchem es sich vom Wort  $s_0$  unterscheidet.

Bestimmen wir z.B. die Unterscheidungskomponente des Wortes  $s_6$ !

$$s_5 = 01110100$$

$$s_6 = 01111000$$

Bilden wir das Komplement von  $s_5$  und führen wir die logische Multiplikation durch !

laufende Nummer	8	7	6	5	4	3	2	1
$\bar{s}_5$	1	0	0	0	1	0	1	1
$s_6$	0	1	1	1	1	0	0	0
$s_6 \wedge \bar{s}_5 = b_6$	0	0	0	0	1	0	0	0

Die Unterscheidungskomponente des Wortes  $s_6$  ist daher 4, weil im Wort  $b_6$  die erste Eins in der vierten Stelle - vom Wortende gerechnet - steht.

Die Unterscheidungskomponenten der obigen Wörter  $s_i$  betragen der Reihe nach

$$6, 5, 7, 6, 5, 4, 8, 7$$

Es sei

$$X = 01110100$$



Die Adressen U und V werden während des Algorithmus mit der Unterscheidungskomponenten bzw. mit der laufenden Nummer des entsprechenden Wortes besetzt. Die Adresse U enthält die Unterscheidungskomponente, wenn im obigen Sinne

$$A_i \notin X$$

Wenn aber

$$A_i \in X,$$

dann wird die Adresse V mit der laufenden Nummer des betreffenden Wortes besetzt.

Der Inhalt der Adressen U und V verändert sich während des Algorithmus mehrmals. Wir geben die nacheinanderfolgenden Inhalte dieser Adressen an

$$U = 0, 0, 0, 0, 0, 0, 4, 8$$

$$V = 0, 1, 2, 3, 4, 5$$

Der letzte Inhalt der Adresse V ergibt die laufende Nummer des gesuchten Wortes X, d.h. in diesem Fall 5 ( $\eta = 5$ ).

Das gesuchte Wort ist daher  $s_5$  :

$$X = s_5$$

### Bemerkungen

a) Im Falle eines geringeren Wortschatzes lohnt es sich nicht, die Wörter nach Gruppen wie oben aufzuteilen. Es genügt, wenn man 4 Gruppen bildet, je nachdem, ob die Wörter eine, zwei, drei oder vier Adressen besetzen.

b) Unser Algorithmus bezieht sich nur auf Wortstämme. Die Endsilben (Endungen) müssen in ihrer Gänze gespeichert werden.

c) In der Praxis haben wir die Wörter in umgekehrter Reihenfolge eingegeben. Das ändert am vorgeführten Algorithmus nichts, hat aber seine Vorteile bei der Analyse der Endungen.

d) Wollen wir nicht das russische Wort im Speicher aufsuchen, sondern gleich das entsprechende ungarische Wort, so braucht man nur in der entsprechenden Gruppe die entsprechende Unterscheidungskomponente aufzusuchen. Wie erwähnt, erhält die erste Adresse die Adresse des entsprechenden ungarischen Wortes.

## L I T E R A T U R

- (1) LOCKE, W.N. - BOTH, A.D.: Machine Translation of Languages, Cambridge (Mass.), 1955.
- (2) COOPER, W.S.: The storage problem, Mechanical Translation, 5/2, 1958.
- (3) Die Rechenanlage M-3 ist eine Zwei-Adressen Maschine. Die Darstellung des Wortes erfolgt in der Gestalt einer Festkommazahl. Das Wort besteht aus 31 Dualstellen (die erste Stelle ist für das Vorzeichen vorbehalten). Ein Magnetkernspeicher fasst 1300 Wörter, ein schnell ansprechbarer Magnetkernspeicher nimmt 1024 Wörter auf. Die Operationszeit beträgt im Durchschnitt 30 Op./sec im Magnettrommelspeicher und 1500 Op./sec im Magnetkernspeicher. Die Ein- und Ausgabe der Informationen erfolgt durch einen Fernschreiber.
- (4) OETTINGER, A.G.: Automatic Language Translation: Lexical and Technical Aspects, Harvard University Press (1960), S. 222.



## ALGORITHMUS FÜR DIE ANALYSE VON ENDUNGEN

F. Kiefer

Der vorliegende Algorithmus beruht auf einem Verfahren, das von B. Dömölki ausgearbeitet wurde. Da unsere Erläuterungen nur in Kenntnis von Dömölki's Artikel verstanden werden können, bringen wir im Anhang eine kurze Zusammenfassung desselben.

Auf Grund des Algorithmus von G. Hell bilden wir vorerst aus den verschiedenen Endungen sogenannte Tabellen und setzen die zusammengehörenden Endungen in Klammer. Haben wir z.B. die Tabelle

$$\left( \begin{array}{c} \bar{B} \\ \bar{T} \quad \text{и} \quad \bar{Й} \\ \bar{Ш} \end{array} \right)$$

so heisst das, dass die Endung "ий" vorliegt, aber vor dem "ий" weder  $\bar{B}$ , noch  $\bar{T}$ , noch  $\bar{Ш}$  steht, Oder die Tabelle

$$\left( \begin{array}{c} e \\ \quad \text{м} \quad y \\ 0 \end{array} \right)$$

bedeutet, dass entweder die Endung "e м y" oder "o м y" vorliegt. Die Tabelle

$$\left( \begin{array}{c} \bar{a} \\ \bar{и} \\ \bar{ы} \\ \bar{я} \end{array} \quad \text{X} \right)$$

bedeutet, dass die Endung  $x$  vorliegt, aber vor  $x$  weder  $a$ , noch  $и$ , noch  $ы$ , noch  $я$  steht.

Wir ordnen diese Tabellen in Anlehnung an den Hellschen Algorithmus den Operationen

$$A_1, A_2, \dots, A_{48}$$

zu.

$$A/1. \quad (a)$$

$$A/2 \quad \begin{pmatrix} \bar{b} \\ \bar{b} \end{pmatrix} \quad (в ш и) \quad (\bar{c} \bar{t} \bar{b}) \quad (c t)$$

$$A/3. \quad \begin{pmatrix} e & \bar{b} \\ o & \end{pmatrix}$$

$$A/4. \quad \begin{pmatrix} a & \bar{m} \\ я & \end{pmatrix}$$

$$A/5. \quad (e \ m)$$

$$A/6. \quad (ы \ m)$$

$$A/7. \quad (o \ m)$$

$$A/8. \quad (и \ m)$$

$$A/9. \quad (e \ й)$$

$$A/10. \quad (o \ й)$$

$$A/11. \quad (ы \ й) \quad \begin{pmatrix} \bar{b} & & & \\ \bar{t} & & & \\ \bar{ш} & & & \end{pmatrix} \quad и \quad й \quad \begin{pmatrix} \bar{e} & \bar{н} & & \\ \bar{я} & \bar{н} & & \end{pmatrix} \quad и \quad й$$

A/12.  $\begin{pmatrix} \bar{в} & & \\ \bar{т} & и & й \\ \bar{д} & & \end{pmatrix} \begin{pmatrix} е & & \\ & н & и \\ & & й \end{pmatrix}$

A/13.  $\begin{pmatrix} е & & \\ о & м & у \end{pmatrix}$

A/14.  $\begin{pmatrix} \bar{е} & \bar{м} & у \\ \bar{о} & \bar{м} & \end{pmatrix}$

A/15.  $\begin{pmatrix} у & \\ ю & ю \end{pmatrix}$

A/16.  $(\bar{ь} \quad ю)$

A/18.  $\begin{pmatrix} \bar{у} & \\ \bar{ь} & ю \\ \bar{ю} & \end{pmatrix}$

A/19.  $(е \quad е)$

A/20.  $(о \quad е)$

A/21.  $(\bar{ы} \quad е) \begin{pmatrix} \bar{в} & & \\ & и & е \\ \bar{т} & & \end{pmatrix} \begin{pmatrix} \bar{а} & \bar{н} & & \\ \bar{е} & \bar{н} & и & е \\ \bar{я} & \bar{н} & & \end{pmatrix}$

A/22.  $\begin{pmatrix} е & & \\ и & т & е \end{pmatrix}$

A/23.  $\begin{pmatrix} \bar{т} & & \\ \bar{в} & и & е \end{pmatrix} \begin{pmatrix} е & & \\ а & н & и \\ я & & е \end{pmatrix}$

A/24.  $\begin{pmatrix} \text{а} & & \text{и} \\ \text{я} & \text{м} & \end{pmatrix}$

A/25.  $\begin{pmatrix} \text{и} & & \text{и} \\ \text{ы} & \text{м} & \end{pmatrix}$

A/26.  $\begin{pmatrix} \text{о} & & \text{о} \\ \text{е} & \text{г} & \end{pmatrix}$

A/27.  $\begin{pmatrix} \bar{\text{о}} & & \text{о} \\ \bar{\text{е}} & \text{г} & \end{pmatrix}$

A/28.  $\begin{pmatrix} \text{а} & & \text{х} \\ \text{я} & & \end{pmatrix}$

A/29.  $\begin{pmatrix} \text{и} & & \text{х} \\ \text{ы} & & \end{pmatrix}$

A/30.  $\begin{pmatrix} \text{а} & & \text{я} \\ \text{я} & & \end{pmatrix}$

A/31.  $(\text{я})$

A/32.  $(\text{ы})$

A/33.  $(\bar{\text{о}} \quad \bar{\text{с}} \quad \text{т} \quad \text{ь})$

A/34.  $(\text{е} \quad \text{ш} \quad \text{ь})$

A/35.  $(\text{и} \quad \text{ш} \quad \text{ь})$

A/36.  $\begin{pmatrix} \bar{\text{е}} & & \text{ь} \\ \bar{\text{и}} & \bar{\text{ш}} & \end{pmatrix} \quad (\text{о} \quad \text{с} \quad \text{т} \quad \text{ь})$



A/37.  $\begin{pmatrix} \text{а} & \text{т} \\ \text{я} & \end{pmatrix}$

A/38.  $\begin{pmatrix} \text{е} & \text{т} \\ \text{и} & \end{pmatrix}$

A/39.  $\begin{pmatrix} \text{у} & \text{т} \\ \text{ю} & \end{pmatrix}$

A/40.  $\begin{pmatrix} \bar{\text{в}} & \bar{\text{ш}} & \text{и} \end{pmatrix} \quad \begin{pmatrix} \bar{\text{а}} \\ \bar{\text{и}} \\ \bar{\text{ы}} \\ \bar{\text{я}} \end{pmatrix} \quad \begin{pmatrix} \bar{\text{м}} & \text{и} \end{pmatrix}$

A/41.  $\begin{pmatrix} \text{с} & \text{ь} \end{pmatrix} \quad \begin{pmatrix} \text{с} & \text{я} \end{pmatrix}$

A/42.  $\begin{pmatrix} \text{с} & \text{т} & \text{в} \end{pmatrix} \quad \begin{pmatrix} \bar{\text{а}} \\ \bar{\text{е}} \\ \bar{\text{и}} \\ \bar{\text{о}} \\ \bar{\text{ы}} \\ \bar{\text{я}} \end{pmatrix} \quad \begin{pmatrix} \bar{\text{а}} \\ \bar{\text{е}} \\ \bar{\text{и}} \\ \bar{\text{у}} \\ \bar{\text{ю}} \\ \bar{\text{я}} \end{pmatrix} \quad \begin{pmatrix} \text{т} \end{pmatrix}$

$\begin{pmatrix} \bar{\text{а}} \\ \bar{\text{и}} \\ \bar{\text{ы}} \\ \bar{\text{я}} \end{pmatrix} \quad \begin{pmatrix} \text{х} \end{pmatrix}$

Die auf diese Weise entstehenden Gruppen haben folgende Bedeutung: Liegt eine bestimmte Endung vor, die durch eine der Tabellen angegeben wird, so sagen wir, dass die Eigenschaft, die die entsprechende Tabelle darstellt, in Erfüllung geht. Geht nun eine der genannten Eigenschaften in Erfüllung, so soll eine weitere Operation durchgeführt werden. Betrachten wir nun z.B. die Gruppe A/21, in dieser Gruppe stehen drei Tabellen, d.h. drei verschiedene Eigenschaften. Geht eine von diesen in Erfüllung, so ist die nächste Operation diejenige, die im Hellschen Algorithmus mit A/21 bezeichnet wurde. Es kann daher vorkommen, dass im Falle der Erfüllung von mehreren Eigenschaften ein und dieselbe Operation durchgeführt wird.

Bilden wir nun aus diesen Gruppen (Tabellen) folgende Tabellen, die wir mit

$$T_1, T_2, \dots, T_k$$

bezeichnen, unter Berücksichtigung folgender Gesichtspunkte:

a) Da wir keinen kyrillischen Fernschreiber besitzen, mussten wir statt der kyrillischen Schrift die lateinische Transkription derselben gebrauchen. In dieser Umschrift mussten wir aber auch weitere Zeichen für kyrillische Buchstaben einführen, da es mehr kyrillische Buchstaben als lateinische gibt. Wir achteten allerdings darauf, dass der umgeschriebene Text lesbar bleibt und dass einer jeden kyrillischen Buchstaben ein Zeichen (lateinischer Buchstabe oder Zahl) entspreche. So wurden z.B. folgende "Zuordnung" getroffen

ч → 4

ь → 6

я → 9

b) Die Negation kann ausser acht gelassen werden. So ergeben z.B. die erste und dritte Tabelle der Gruppe der Gruppe A/2 dieselbe Tabelle

( v )

Wir erhalten nun unter Berücksichtigung von Punkt a) und b) folgende Tabellen:

$$T_1 = (a)$$

$$T_2 = (v)$$

$$T_3 = (v \text{ s } i)$$

$$T_4 = (2 \text{ t})$$

$$T_6 = \begin{pmatrix} e & v \\ o & \end{pmatrix}$$

$$T_6 = \begin{pmatrix} a & m \\ 9 & \end{pmatrix}$$

$$T_7 = (e \text{ m})$$

$$T_8 = (y \text{ m})$$

$$T_9 = (o \text{ m})$$

$$T_{10} = (i \text{ m})$$

$$T_{11} = (e \text{ j})$$

$$T_{12} = (o \text{ j})$$

$$T_{13} = (y \text{ j})$$

$$T_{14} = (i \text{ j})$$

$$T_{15} = \begin{pmatrix} v & & \\ t & i & j \\ c & & \end{pmatrix}$$

$$T_{16} = \begin{pmatrix} e & & & \\ 9 & n & i & j \end{pmatrix}$$

$$T_{17} = \begin{pmatrix} e & & \\ o & m & u \end{pmatrix}$$

$$T_{18} = (u)$$

$$T_{19} = \begin{pmatrix} u & \\ q & q \end{pmatrix}$$

$$T_{20} = (6 \text{ q})$$

$T_{21}$	=	(q)	$T_{22}$	=	(e e)
$T_{23}$	=	(o e)	$T_{24}$	=	(y e)
$T_{25}$	=	(i e)	$T_{26}$	=	$\begin{pmatrix} e & t & e \\ i \end{pmatrix}$
$T_{27}$	=	$\begin{pmatrix} t & i & e \\ v \end{pmatrix}$	$T_{28}$	=	$\begin{pmatrix} e & & & \\ a & n & i & e \\ 9 \end{pmatrix}$
$T_{29}$	=	$\begin{pmatrix} a & m & i \\ 9 \end{pmatrix}$	$T_{30}$	=	$\begin{pmatrix} i & m & i \\ y \end{pmatrix}$
$T_{31}$	=	$\begin{pmatrix} o & g & o \\ e \end{pmatrix}$	$T_{32}$	=	(o)
$T_{33}$	=	$\begin{pmatrix} a & h \\ 9 \end{pmatrix}$	$T_{34}$	=	$\begin{pmatrix} i & h \\ y \end{pmatrix}$
$T_{35}$	=	$\begin{pmatrix} a & 9 \\ 9 \end{pmatrix}$	$T_{36}$	=	(9)
$T_{37}$	=	(y)	$T_{38}$	=	(t 6)
$T_{39}$	=	(e s 6)	$T_{40}$	=	(i s 6)
$T_{41}$	=	(6)	$T_{42}$	=	(o 2 t 6)
$T_{43}$	=	$\begin{pmatrix} a & t \\ 9 \end{pmatrix}$	$T_{44}$	=	$\begin{pmatrix} e & t \\ i \end{pmatrix}$
$T_{45}$	=	$\begin{pmatrix} u & t \\ q \end{pmatrix}$	$T_{46}$	=	(i)

$$T_{47} = (2 \ t \ v)$$

$$T_{48} = (m)$$

$$T_{49} = (t)$$

$$T_{50} = (h)$$

$$T_{51} = (2 \ 6)$$

$$T_{52} = (2 \ 9)$$

Wir teilen nun diese Tabellen in weitere Gruppen ein, je nachdem, welcher Buchstabe in der letzten Spalte der Tabelle steht (welche der letzte Buchstabe des Wortes bzw. der Endung ist) und erhalten folgende Gruppen:

$$a: T_1$$

$$v: T_5, T_{47}, T_2$$

$$e: T_{27}, T_{28}, T_{22}, T_{23}, T_{24}, T_{25}, T_{26}$$

$$i: T_3, T_{29}, T_{30}, T_{46}$$

$$j: T_{15}, T_{16}, T_{11}, T_{12}, T_{13}, T_{14}$$

$$m: T_6, T_7, T_8, T_9, T_{10}, T_{48}$$

$$o: T_{31}, T_{32}$$

$$t: T_4, T_{43}, T_{44}, T_{45}, T_{49}$$

$$u: T_{17}, T_{18}$$

$$h: T_{33}, T_{34}, T_{50}$$

$$y: T_{37}$$

$$6: T_{42}, T_{38}, T_{39}, T_{40}, T_{51}, T_{41}$$

$$q: T_{19}, T_{20}, T_{21}$$

$$9: T_{35}, T_{52}, T_{36}$$

Es handelt sich nun darum, dass man diese Tabellen irgendwie im Speicher unterbringen muss. Das geschieht unter Berücksichtigung folgender Gesichtspunkte:

a) Man gruppiert die Tabellen  $T_i$  so, dass die entsprechenden Reihenvektoren in einem Wort untergebracht werden können. Das heisst, bezeichnet man die Zahlen der Spalten der einzelnen Tabellen mit

$$p_1, p_2, \dots, p_t$$

so darf keinesfalls in einer Gruppe die Summe

$$p_1 + p_2 + \dots + p_s, \quad \text{wo } s \leq t,$$

grösser sein als die Zahl der Dualstellen in einem Wort, d.h. als 30 (im Achtersystem 36!).

b) Die Tabellen werden in umgekehrter Reihenfolge eingespeichert. Das heisst, dass zuerst die letzte Spalte kommt, dann folgt die vorletzte usw. Eine geeignete Gruppierung kann zur Folge haben, dass man schon vom ersten Buchstaben entscheiden kann, ob die gesuchte Endung in jene Gruppe gehört oder nicht.

c) Bei der Gruppierung kann man auch noch einen statistischen Gesichtspunkt berücksichtigen: Man soll möglichst die häufigeren Endungen in die ersten Gruppe reihen.

Wir unterbringen nun zuerst diejenigen Tabellen, deren letztes Element j oder 6 ist. Die Zahl der Spalten  $\sum p_i$  beträgt jetzt eben 30; die einzelnen Reihenvektoren bleiben innerhalb eines Wortes.

In die zweite Gruppe gehören diejenigen Tabellen, deren letztes Element e oder i ist, diese Tabellen ergibt die Spaltenzahl 28.

Die dritte Gruppe bilden diejenigen Tabellen, deren letztes Element m, t oder v ist. Die Zahl der Spalten beträgt 26.

In die vierte und letzte Gruppe gehören diejenigen Tabellen, deren letztes Element a, u, q, o, h, 9 oder y ist. Die Zahl der Spalten beträgt 25.

Haben wir nun ein unbekanntes Wort  $X$ , so ist zuerst die Frage zu entscheiden, welcher Buchstabe am Ende des Wortes steht. (Natürlich nur dann, wenn das Wort nicht im "Wörterbuch" aufgefunden wurde.) Erst wenn diese Frage entschieden ist, kann der Algorithmus weitergeführt werden.

Auf Grund der einzelnen Gruppen verfertigen wir eine Matrix, die wir mit  $B$  bezeichnen, folgenderweise:

Jedem Buchstaben, der in der betreffenden Gruppe vorkommt, ordnen wir einen Reihenvektor zu. Enthält die Gruppe über  $q$  verschiedene Buchstaben, so hat die Matrix genau  $q$  Reihen. Die Zahl der Spalten wurde schon bei den einzelnen Gruppen bestimmt. Jeder Reihenvektor enthält dort und nur dort eine Eins, wo der betreffende Buchstabe in einer zur Gruppe gehörender Tabelle vorkommt. In anderen Stellen steht Null.

Wir bezeichnen mit  $K$  und  $V$  diejenigen Vektoren, die dazu dienen, um die einzelnen Tabellen voneinander trennen zu können. Der Vektor  $K$  enthält Eins an den Anfangsspalten der einzelnen Tabellen, der Vektor  $V$  aber an den Endspalten derselben.

Tafel I. enthält folgende Tabellen

$T_{11}, T_{12}, T_{13}, T_{14}, T_{15}, T_{16}, T_{42}, T_{38}, T_{39}, T_{40},$   
 $T_{51}, T_{41};$

Tafel II. enthält die Tabellen

$T_{22}, T_{23}, T_{24}, T_{25}, T_{26}, T_{27}, T_{28}, T_3, T_{29}, T_{30}, T_{46},$

Tafel III. enthält die Tabellen

$T_6, T_7, T_8, T_9, T_{10}, T_{48}, T_4, T_{43}, T_{44}, T_{45},$   
 $T_{49}, T_5, T_{47}, T_2$

Tafel IV. enthält folgende Tabellen

$T_1, T_{17}, T_{18}, T_{19}, T_{20}, T_{21}, T_{31}, T_{32}, T_{33}, T_{34},$   
 $T_{50}, T_{35}, T_{52}, T_{36}, T_{37}.$

Die einzelnen Reihenvektoren sind auch im Achtersystem

IM ACHTERSYSTEM

I M D U A L S Y S T E M

		1	2	3	4	5	6	7	10	11	12	13	14	15	16	17	20	21	22	23	24	25	26	27	30	31	32	33	34	35	36	
K <sub>0</sub>	52 5104 2445	1	0	1	0	1	0	1	0	1	0	1	0	0	0	1	0	0	0	1	0	1	0	0	1	0	0	1	0	1	0	1
V <sub>0</sub>	25 2210 5113	0	1	0	1	0	1	0	1	0	0	1	0	0	0	1	0	0	0	1	0	1	0	0	1	0	0	1	0	1	1	
BUCHST.	////	////	////	////	////	////	////	////	////	////	////	////	////	////	////	////	////	////	////	////	////	////	////	////	////	////	////	////	////	////	////	
j	52 5100 0000	1	0	1	0	1	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
e	20 0010 0100	0	1	0																												
o	04 0000 4000				1																											
y	01 0000 0000					1																										
i	00 2440 0010									1		1			1																1	
v	00 0200 0000																															
t	00 0202 1000																															
c	00 0200 0000																															
n	00 0020 0000																															
q	00 0010 0000																															
6	00 0004 2445																															
2	00 0001 0002																															
s	00 0000 0220																															

Die Matrix B besteht aus 13 Reihen und 30 Spalten.



IM ACHTERSYSTEM

IM DUALSYSTEM

		1	2	3	4	5	6	7	10	11	12	13	14	15	16	17	20	21	22	23	24	25	26	27	30	31	32	33	34	35	36						
K <sub>1</sub>	52 5110 4444	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1						
V <sub>1</sub>	25 2221 1114	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1						
BÜCHST.	//////	//////																																			
e	72 5311 0000	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1						
o	04 0000 0000			1																																	
y	01 0000 0010					1																															
i	00 2244 4454								1			1		1		1		1		1		1		1		1		1		1							
t	00 0420 0000									1					1																						
v	00 0020 1000															1								1													
n	00 0002 0000																																				
a	00 0001 0100																																				
q	00 0001 0100																																				
s	00 0000 2000																																				
m	00 0000 0220																																				

Die Matrix B besteht aus 11 Reihen und 28 Spalten.

515

## IM ACHTERSYSTEM

## IM DUALSYSTEM

		1	2	3	4	5	6	7	10	11	12	13	14	15	16-17	20	21	22	23	24	25	26	27	30	31	32	33	34	35	36
K <sub>2</sub>	52 5325 3220	1	1		1	1			1		1	1		1		1		1	1	1			1			1				
V <sub>2</sub>	25 2652 6460		1	1	1		1			1	1		1		1		1		1	1			1			1	1			
BUCHST.	/ /																													
m	52 5200 0000	1	1		1	1			1		1																			
a	20 0010 0000		1												1															
q	20 0010 0000		1												1															
e	04 0002 0400				1											1							1							
y	01 0000 0000					1																								
o	00 0000 0400																							1						
u	00 0400 4000										1										1									
t	00 0126 2100											1		1		1		1		1					1					
2	00 0040 0040														1												1			
i	00 2002 0000								1								1													
q	00 0000 4000																			1										
v	00 0000 1220																						1		1			1		

Die Matrix B besteht aus 12 Reihen und 26 Spalten.

IM ACHTERSYSTEM

IM DUALSYSTEM

		1	2	3	4	5	6	7	10	11	12	13	14	15	16	17	20	21	22	23	24	25	26	27	30	31	32	33	34	35	36					
K <sub>3</sub>	63 2632 6540	1	1			1	1		1		1	1			1	1		1		1	1		1		1		1									
V <sub>3</sub>	46 5465 5340	1			1	1	1		1	1				1	1		1		1	1		1		1	1	1										
BUCHST.																																				
a	40 0004 1000	1				1		1							1								1													
u	22 4000 0000		1																																	
m	10 0000 0000					1																														
e	04 0040 0000					1									1																					
o	04 0260 0000					1						1			1	1																				
q	01 6400 0000							1	1	1		1																								
6	00 1000 0000										1																									
g	00 0100 0000												1																							
h	00 0012 4000																1		1		1															
9	00 0004 3500																1					1	1	1		1										
i	00 0001 0000																				1															
y	00 0001 0040																				1										1					
2	00 0000 0200																																	1		

Die Matrix B besteht aus 13 Reihen und 25 Spalten

An Hand dieser Tafeln kann die Analyse einer Endung folgendermassen erfolgen: Nehmen wir an, es soll das Wort "сталом" analysiert werden. Nach der entsprechenden Transkription und der Umkehrung erhalten wir das Wort molot2. So wird dieses Wort unter den Wortstämmen nicht aufgefunden, da dort nur lot2 steht. Es muss daher die Frage entschieden werden, welche Eigenschaft der Tafeln I-IV. in Erfüllung geht. Das wird aber durch die Formel.

$$Q_0 = \underline{0}$$

$$Q_k = \left\{ \left[ \begin{array}{cc} 0_{k-1} & \bar{V} \end{array} \right] \vee K \right\} \wedge B_{S_k}$$

bestimmt wo  $\underline{0}$  den Nullvektor bedeutet, " $\wedge$ " und " $\vee$ " sind Zeichen der Konjunktion und Disjunktion,  $\bar{V}$  bedeutet das Komplement von  $V$ , die Multiplikation mit  $2^{-1}$  bedeutet eine Rechtsverschiebung,  $B_{S_k}$  bezeichnet den entsprechenden Reihenvektor der Matrix  $B$ .

Da der erste Buchstabe des bestimmenden Wortes  $m$  ist, braucht man die obigen Operationen nur mit der dritten Gruppe (Tafel III.) durchzuführen.

$$Q_0 = 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000$$

Die Konjunktion von  $Q_0$  mit  $V$  ergibt jetzt den Nullvektor, nach Rechtsverschiebung bleibt der Nullvektor unverändert, die Disjunktion mit  $K$  ergibt den Vektor  $K$ . Das heisst, bei der Bildung von  $Q_1$  braucht man jetzt nur die Konjunktion des Vektors  $K$  mit dem entsprechenden Vektor  $B_m$  (mit  $B_m$  bezeichnen wir den zum Buchstaben  $m$  gehörenden Reihenvektor der Matrix  $B$ ) bilden.

$$K = 101\ 010\ 101\ 011\ 010\ 101\ 011\ 010\ 010\ 000$$

$$B_m = 101\ 010\ 101\ 010\ 000\ 000\ 000\ 000\ 000\ 000$$

$$Q_1 = K \wedge B_m = \underline{101\ 010\ 101\ 010\ 000\ 000\ 000\ 000\ 000\ 000}$$

Eine notwendige und hinreichende Bedingung dafür, dass eine Eigenschaft in Erfüllung geht, ist, dass

$$Q_k \wedge V \neq 0$$

Nun ist aber

$$Q_1 = 101\ 010\ 101\ 010\ 000\ 000\ 000\ 000\ 000\ 000$$

$$V = 010\ 101\ 010\ 110\ 101\ 010\ 110\ 100\ 110\ 000$$

$$Q_1 \wedge V = 000\ 000\ 000\ 010\ 000\ 000\ 000\ 000\ 000\ 000$$

Est ist also

$$Q_1 \wedge V \neq 0$$

Man muss als nächstes feststellen, welche Eigenschaft in Erfüllung gegangen ist.

Die Spaltenlänge der einzelnen Tabellen, die zur Gruppe III. gehören, beträgt der Reihe nach

$$2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 1, 2, 3, 1$$

Im Vektor  $Q_1 \wedge V$  steht aber in der 11-ten Stelle Eins, d.h. in der

$$2 + 2 + 2 + 2 + 2 + 1 = 11$$

-ten Stelle. Das bedeutet aber, dass die der 6-ten Tabelle entsprechende Eigenschaft in Erfüllung ging.

Es wird aber offensichtlich nach der Trennung von  $m$  zurückgebliebene Wort  $olot2$  auch nicht unter den Wortstämmen aufgefunden. Das Verfahren muss daher fortgesetzt werden.

$$Q_1 = 101\ 010\ 101\ 010\ 000\ 000\ 000\ 000\ 000\ 000$$

$$V = 101\ 010\ 101\ 001\ 010\ 101\ 001\ 011\ 001\ 111$$

$$Q_1 \wedge V = 101\ 010\ 101\ 000\ 000\ 000\ 000\ 000\ 000\ 000$$

Nach der Rechtsverschiebung um eine Stelle

$$Z = 010\ 101\ 010\ 100\ 000\ 000\ 000\ 000\ 000\ 000$$

$$K = 101\ 010\ 101\ 011\ 010\ 101\ 011\ 010\ 010\ 000$$

$$Z \vee K = 111\ 111\ 111\ 111\ 010\ 101\ 011\ 010\ 010\ 000$$

Der Reihenvektor  $B_0$ , der dem Buchstaben o entspricht, ist nun

$$B_0 = 000\ 000\ 010\ 000\ 000\ 000\ 000\ 000\ 100\ 000\ 000$$

$$Z \vee K = 111\ 111\ 111\ 111\ 010\ 101\ 011\ 010\ 010\ 000$$

$$(Z \wedge K) \vee B_0 = Q_2 = 000\ 000\ 010\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000$$

und die Konjunktion mit V ergibt

$$Q_2 \wedge V = 000\ 000\ 010\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000$$

das heisst

$$Q_2 \wedge V \neq 0$$

und im Vektor  $Q_2 \wedge V$  steht in der achten Stelle Eins: da

$$2 + 2 + 2 + 2 = 8 \text{ ist,}$$

so ist die Eigenschaft, die der vierten Tabelle entspricht, in Erfüllung gegangen. Nach der Abtrennung des Buchstabens o, bleibt nun das Wort lot2 zurück, dass schon unter den Wortstämmen aufgefunden werden kann.

Wenn nun das Wort noch immer nicht unter den Wortstämmen auffindbar wäre, so müsste man das Verfahren wieder fortsetzen. In der Bildung von  $Q_3$  müsste man jetzt aber den Reihenvektor nehmen, der dem Buchstaben l entspricht, solcher ist aber in der Matrix der 2. Gruppe nicht zu finden. Jetzt wissen wir schon, dass das Wort lot2 nicht im "Wörterbuch" vorhanden ist. Auch wenn wir den entsprechenden Reihenvektor fänden, würde es sich bald herausstellen, dass keine der Eigenschaften in Erfüllung geht und so weiss man wieder, dass das Wort nicht auffindbar ist.

Das Wort molot2 erfüllt also die Eigenschaft, die durch die Tabellen  $T_{48}$  und  $T_9$  angegeben ist. Wir sagen auch, dass das Wort molot2 von Typ  $T_{48}$  und  $T_9$  ist. Was nun die Abtrennung der Endung anbetrifft, so kommt immer die stärkere Eigenschaft in Betracht. Man soll daher das Verfahren fortsetzen, bis man das Wort (den Wortstamm) findet (oder nicht findet) und die

letzte erfüllte Eigenschaft ist die Stärkste. Man wird daher auch nicht die mit A/42, sondern die mit A/7 bezeichnete Operation durchführen.

Die Abtrennung der Endungen geschieht durch logische Multiplikation. Nicht in allen Fällen darf aber die ganze Endung (d.h. der Länge der Tabelle entsprechenden Buchstabenanzahl) abgetrennt werden.

Im Falle der Tabellen

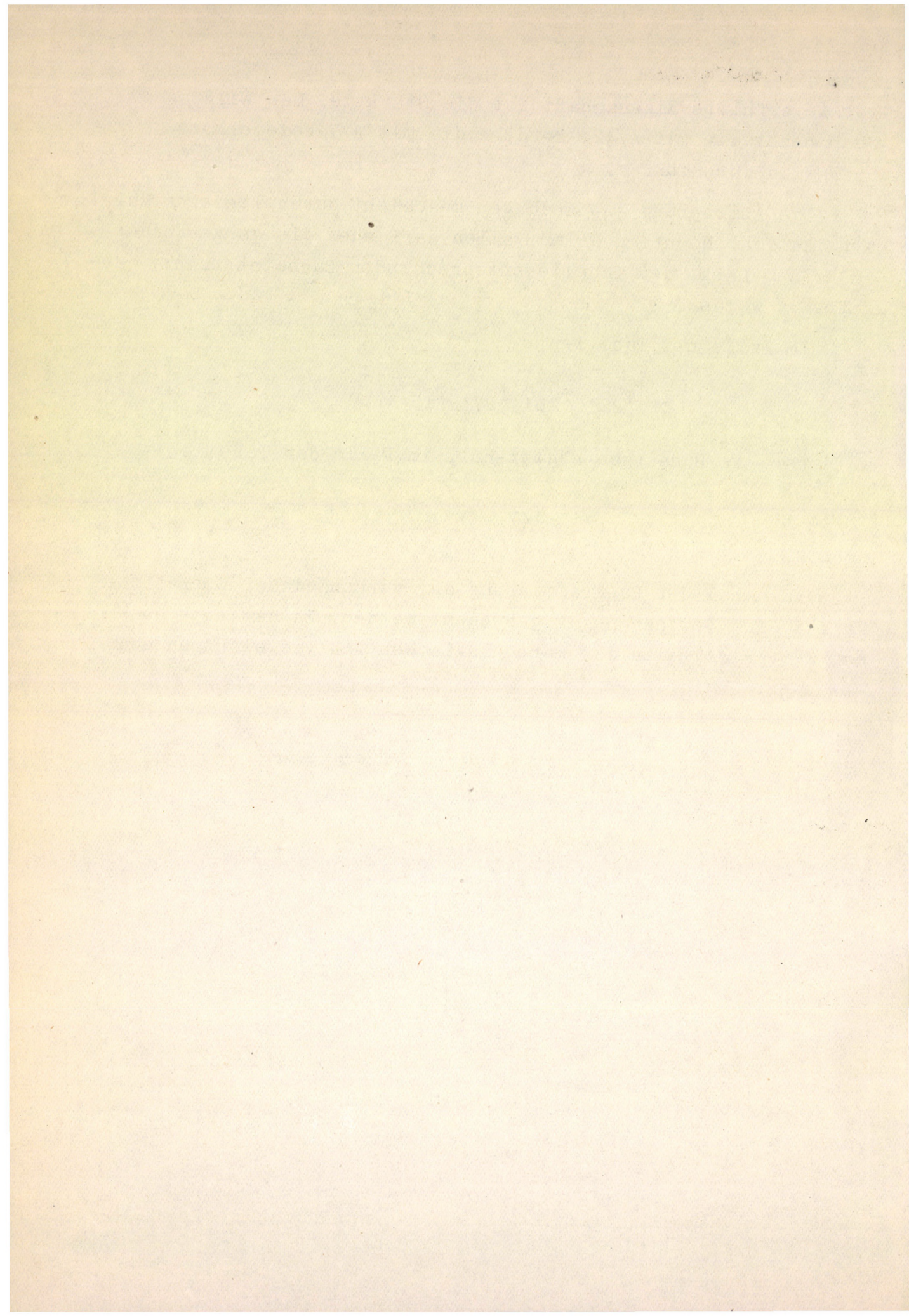
$T_{28}, T_{16}, T_{27}, T_{15}$

wird nur ein Buchstabe abgetrennt, im Falle der Tabelle

$T_{47}$

nichts.

Auch bei der Programmierung der verschiedenen Operationen kann dieses Verfahren gut gebraucht werden. Es hat sich auch bei weiteren Teilen der morphologischen Analyse als brauchbar erwiesen.





# MORPHOLOGICAL ANALYSIS BY HELP OF THE METHOD OF SUCCESSIVE DELIMITATION

By Dénes Varga

## INTRODUCTION

Hereinafter I would expound a process with help of which the morphologic and partly syntactic analysis can be made in a substantially simpler and quicker way in the case of strongly inflected languages /as for instance the Russian/, than by processes to the best of our knowledge applied up to the present, even if it does not cover all details of analysis. This process can be programmed with relatively few commands, so that it may be applied with success on machines with lesser performance. /The experimental program has already been completed for the computer M-3 of the Computing Centre of the Hungarian Academy of Sciences and it is just about being tested./

## I. CLASSIC METHOD OF MACHINE ANALYSIS AND THE HUMAN ANALYSIS

The usual method of the analysis consists in breaking down the linguistic problems to the sequence of their logic elements of questions to be solved\*. The problem reduced to game "Bar-Kochba" by such a way can already be simplified on the machine to the comparison of the contents of cells, afterwards the analysis may be continued in one or the other direction, depending on the result of the comparison. This method attains quick its goal in a hierarchically construed system, where we have to do with serially superposed categories. But in the case of linguistic analysis where as a rule we have to choose

---

\* Cf. И.А.Мельчук: О машинном переводе с венгерского языка на русский. Проблемы кибернетики, вып.1. стр.231.

out of many juxtaposed categories the adequate one, this process degenerates to mere guesswork. Besides it does not take due advantage of the possibilities of the machine, for the contents of whole cells are always compared even if the examination covers but a sole letter or suffix. A special complication is caused by the frequency of "multi-valued", homonymous suffixes, words, sets of words which could have as a consequence that we may discover only after having completed a long analysis in one direction that we are on the wrong track and we have to continue the analysis on an entirely other way.

Human analysis is much more complex than this: it monitors and coordinates at the same time several points of views, with respect to the Russian language it is true, for instance, that the preposition с may be followed by several cases, by genitive, instrumental and also by accusative /if we consider singular and plural separately, that makes already 6 possibilities/ the possessive pronoun нашей may mean 4 cases: genitive, dative, instrumental or prepositional in the singular, the substantive фабрики may mean genitive singular, nominative or accusative plural, - yet every Russian discovers instantaneously that the expression с нашей фабрики stands in genitive singular; within the given connection of words these words, though having grammatically several meanings, establish unequivocally one possible meaning of each other. [1] Of course, the unambiguousness is not always secured, even then not, if it defines unambiguously the given connection of words. For instance по нашей стороне may mean "on our side" /Идти по нашей стороне /, but also "according to our client" /по нашей стороне дело обстоит не так /. Unambiguousness may be secured in such cases only by words standing around the expression in question.

The problem - solved quasi automatically by those who know the language consists in the examination of the govern-

ment of the preposition, of the compatibility of the pronominal and substantive suffixes within a nominal syntagma.

The same complex examination is undertaken by the knower of the language with respect not only to larger but also smaller unities, often without the process becoming conscious for him.

The ending -ей may cover grammatical categories of many kinds. The principal categories are

- 1/ Genitive feminine singular of adjectives and pronouns
- 2/ Dative " " " " " "
- 3/ Instrumental " " " " " "
- 4/ Prepositive " " " " " "
- 5/ Comparative of adjectives
- 6/ Instrumental singular of feminine substantives
- 7/ Genitive plural " masculine "
- 8/ Genitive plural of certain neutral substantives
- 9/ Accusative plural of substantives meaning male living creatures.

In connection with the words нашей, only those possibilities out of the above enumerated many ones, can be considered, which are compatible with our knowledge concerning the stem наш-. On basis of the co-ordination of our knowledge concerning the stem and the suffix can we decide, that the pronoun нашей may stand in one of the 4 cases enumerated previously. We have not even mentioned here that the на- might be a prefix in the given word, and -шей the imperative of the verb шить /to sew/: for instance "нашей пуговицу!" "Sew the button on!" It is outside the frame of the word to decide what is stem in the given case /if we consider the text written down without accent/.

Similarly it can be stated that the ending -и may mean

singular genitive, dative, prepositive of feminine substantives, plural nominative and accusative of masculine, feminine, furthermore certain neuter substantives, plural of the shortened form of adjectives, imperative of verbs etc., yet when it is connected with the stem фабрик the number of the possibilities is substantially reduced.

In the case of single suffixes the problem can be reduced to the examination of the compatibility of letters within the suffix, resp. of the compatibility of informations referring to the suffix and stem. In such a conception, we have to handle a problem which is of the same type as in the previous case.

## II. THE METHOD OF SUCCESSIVE DELIMITATION

Instead of the "Bar-Kochba"-like analysis a method, standing nearer to the analytic faculty of man, can be designed for electronic computers by applying the process of "logical multiplication" [2]. This process corresponds in the binary system to the multiplication by bits /digits/: the logic multiplication of two numbers contains 1 on the place of those bits, on the place of which 1 stood in both numbers. This process enables us to establish the common part of sets, if the elements of the set is represented by "1"-s standing on the place of the single bits. Our task consists in reducing the problem of the compatibility of certain qualities to the establishment of the common part of sets - and then we can do it on the machine very simply, with a sole operation, at the sametime for all bits.

Let us make correspond serially to each of the bits of the electronic computer a grammatical category, such categories we wish to examine simultaneously, for instance different cases when examining a noun, different forms when examining a verb etc. Say nominative singular must correspond to bit 1., accusative singular to bit 2., genitive singular to bit 3., etc. Let us consider the aggregate of these qualities as basic set, Now if we let correspond to single words of a nominal syntagma "numbers" /more precisely : Boole-vectors/ containing such 0-s and 1-s, which contain 1 on the place of the categories which could refer to the respective word and 0 on the place of the other categories, then the Boole-vectors so obtained will represent a partial set of the basic set consisting of the totality of the qualities examined simultaneously.

Let us construct the following correspondence as a simplified model between grammatical categories and bits:

Bit 1.	nominative singular	/abbr. N/
" 2.	accusative	" /A/
" 3.	genitive	" /G/
" 4.	dative	" /D/
" 5.	instrumental	" /I/
" 6.	prepositive	" /P/
" 7.	nominative plural	/N/
" 8.	acc.	" /A/
" 9.	gen.	" /G/
" 10.	dat.	" /D/
" 11.	instr.	" /I/
" 12.	prep.	" /P/

In this system the following Boole-vector corresponds to the preposition с /according to its government/. For the sake of perspicuity we indicate, too, the categories corresponding to the single bits:

$$\begin{array}{cccccccccccc}
 & N & A & G & D & I & P & N & A & G & D & I & P \\
 \psi_1 = & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0
 \end{array}$$

The following Boole-vector may be assigned /on basis of the possible cases/ to the pronoun нашей:

$$\begin{array}{cccccccccccc}
 & N & A & G & D & I & P & N & A & G & D & I & P \\
 \psi_2 = & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0
 \end{array}$$

By logical multiplication of these two vectors we obtain 1 on the place of those categories which the words с and нашей simultaneously satisfy, that is where 1 stands in both vectors

on the place of the corresponding bit:

N A G D I P N A G D I P

$$\psi_1 \wedge \psi_2 = 0 0 1 0 1 0 0 0 0 0 0 0$$

Knowing the correspondence of the bits and grammatical categories we can establish from the result of the logical multiplication that within a nominal syntagma these two words are compatible only in the genitive singular and instrumental singular.

The Boole-vector pertaining to the word фабрики will be the following:

N A G D I P N A G D I P

$$\psi_3 = 0 0 1 0 0 0 1 1 0 0 0 0$$

Carrying out a new logical multiplication between the former logical product and  $\psi_3$ , we obtain the following result:

N A G D I P N A G D I P

$$(\psi_1 \wedge \psi_2) \wedge \psi_3 = 0 0 1 0 0 0 0 0 0 0 0 0$$

The result so obtained is already ambiguous in the expression с нашей фабрики pronoun and substantive are standing in genitive singular.

As we can see from the above example, the number of possibilities decreases thanks to logical multiplications performed in sequence, moreover, in lucky cases it may be reduced to a sole case.

The essence of the method consists therefore of looking after the known properties of the object under examination,

one after another, and find out what conceptual categories out of those represented by these properties may be compatible with each other. So we tighten gradually, successively the set of categories which may be referred to the object under examination. Our purpose is to define that smallest subset of the set of all categories, to which the object under examination pertains.

The method may be applied in several scales: the assignment of the conceptual categories to the bits is conditioned, it depends on the character of the object under examination and on the informations at our disposal. In first approximation we may content ourselves with the distinction of the principal categories /for the number of bits at our disposal is limited/, and later, after the adequate narrowing of the set we may make correspond more categories for any single category in a new system, and break down the categories into further subcategories.



### III. EXPOSITION OF THE MODEL OF MACHINE ANALYSIS I.

#### /Preliminary informations/

After the exposition of the applied method let us examine the progress of the analysis on a more complete model from the beginning to the analysis on syntagmatic level.

Before surveying the operation of the model let us consider in so many words the model itself.

We have the computer M-3 at our disposal, where the length of a cell is of 30 bits. To the machine an input unit with perforated tape of 5 channels is joined. The signs on the perforated tape are perforated according to the Telex codes.

#### 1/ Transcription. Text input. Arrangement of the words in the memory

In order of the better utilization of the bits we constructed such a transliteration of the text of Russian letters which uses for the transliteration of the letters only the 26 ones of the Telex alphabet, but not the signs. We attained the reduction of the numbers of the letters of the Russian alphabet the following way:

1/ We have indicated certain letters, used rather infrequently, by a letter of two signs in the transliteration. The indication of the so-called sh-voices /щипящие/ by two letter-signs seemed the most practical. For the double letters we used the same letter /h/ as modifying sign

ж → zh      ч → ch      ш → sh      щ → th.

2/ We have transcribed some Russian letters which have identical or related functions, with the same Roman letter:

Й, Ь, Ъ → j

е, ё, э → e.

The number of the homonymous words increases thereby only very slightly

/ поэт, поёт → poet/.

In transcribing the other letters, we conserved possibly their natural correspondents in the Roman alphabet, according to the internationally used pronunciation:

<u>а</u>	→	a	<u>Н</u>	→	n
<u>б</u>	→	b	<u>о</u>	→	o
<u>в</u>	→	v	<u>п</u>	→	p
<u>г</u>	→	g	<u>р</u>	→	r
<u>д</u>	→	d	<u>с</u>	→	s
<u>з</u>	→	z	<u>т</u>	→	t
<u>и</u>	→	i	<u>у</u>	→	u
<u>к</u>	→	k	<u>ф</u>	→	f
<u>л</u>	→	l	<u>ц</u>	→	c
<u>м</u>	→	m			

Exception is made for x to which we have made to correspond the sign х of the Roman alphabet so that the modifying sign h won't get to functions

х → x,

as well as for ю and я to which we have assigned to letters q and w which are not to be found in the Russian alphabet:

ю    →    w  
я    →    q

For the indication of ы we have used, on the analogy of the Czech and Slovak orthography, the letter y.

The transliteration of the whole Russian alphabet is therefore shown by the following table:

<u>а</u> → a	<u>и</u> → i	<u>р</u> → r	<u>ш</u> → sh
<u>б</u> → b	<u>й</u> → j	<u>с</u> → s	<u>щ</u> → th
<u>в</u> → v	<u>к</u> → k	<u>т</u> → t	<u>ъ</u> → j
<u>г</u> → g	<u>л</u> → l	<u>у</u> → u	<u>ы</u> → y
<u>д</u> → d	<u>м</u> → m	<u>ф</u> → f	<u>ь</u> → j
<u>е, ё</u> → e	<u>н</u> → n	<u>х</u> → x	<u>э</u> → e
<u>ж</u> → zh	<u>о</u> → o	<u>ц</u> → c	<u>ю</u> → w
<u>з</u> → z	<u>п</u> → p	<u>ч</u> → ch	<u>я</u> → q

By such a transcription it is enough, also in the machine, to use 5 bits to indicate single letters. So we may put 6 letters in a cell and even the longest Russian word finds room in 3 cells. The use of the number switch is needed only when punctuation marks and numbers, formulas between the text are written, therefore these can be separated at once from the text to be translated.

The arrangement of a single word is made, according to the length of the word, in 1, 2 or 3 cells /m1, m2, m3/. The first letter of the word occupies the place of the first 5 bits, the second letter that of the next 5 bits and so on,

thereafter the word continues in the m2 cell in a similar way, from left to right. After the last letter we fill the cell which we have begun, but not wholly filled, with 0-s.

We may ease the troublesome work of the preparation of the dictionary part by conserving the Russian alphabetical order in the dictionary. It is advisable therefore the trans-coding of the Telex signs in the Russian alphabetical order according to the following correspondence:

Russian		Roman	in binary system	in octonary system
	space		→ 00000	0
<u>а</u>	→	a	→ 00001	1
<u>б</u>	→	b	→ 00010	2
<u>в</u>	→	v	→ 00011	3
<u>г</u>	→	g	→ 00100	4
<u>д</u>	→	d	→ 00101	5
<u>е, ё</u>	→	e	→ 00110	6
<u>ж</u>	→	zh	→ 0011111010	7+32
<u>з</u>	→	z	→ 00111	7
<u>и</u>	→	i	→ 01000	10
<u>й</u>	→	j	→ 10111	27
<u>к</u>	→	k	→ 01001	11
<u>л</u>	→	l	→ 01010	12
<u>м</u>	→	m	→ 01011	13
<u>н</u>	→	n	→ 01100	14
<u>о</u>	→	o	→ 01101	15
<u>п</u>	→	p	→ 01110	16

Russian		Roman		in binary system	in octonary system
<u>р</u>	→	r	→	01111	17
<u>с</u>	→	s	→	10000	20
<u>т</u>	→	t	→	10001	21
<u>у</u>	→	u	→	10010	22
<u>ф</u>	→	f	→	10011	23
<u>х</u>	→	x	→	10100	24
<u>ц</u>	→	c	→	10101	25
<u>ч</u>	→	ch	→	1010111010	25+32
<u>ш</u>	→	sh	→	1000011010	20+32
<u>щ</u>	→	th	→	1000111010	21+32
<u>ъ</u>	→	j	→	10111	27
<u>ы</u>	→	y	→	10110	26
<u>ь</u>	→	j	→	10111	27
<u>е</u>	→	e	→	00110	6
<u>ю</u>	→	w	→	.11000	30
<u>я</u>	→	q	→	11001	31

On basis of the above described arrangement of the words and of the transcoding according to the Russian alphabet we may interpret the relationship of order of magnitude between the words in such a way that it should essentially correspond to the alphabetical order of words. For instance

staryj < stol

for a < o and

stol < stolom

because space < 0. /Compare to 0.34721 < 0.348 and 0.348 < 0.34813./

resp. with the same numbers:

	s	t	a	r	y	j	
	10000	10001	00001	01111	10110	10111	<
<	s	t	o	l	□	□	
	10000	10001	01101	01010	00000	00000	<
<	s	t	o	l	o	m	
	10000	10001	01101	01010	01101	01011	

Therefore, if we arrange the numbers corresponding to the words of the dictionary in their order of magnitude, then we virtually leave the Russian alphabetical order unchanged. Divergences may arise only from the reduction of the number of letters.

The putting of the modifying sign h on the last place of the alphabet took place with the purpose to avoid the mingling in the dictionary of z and zh, s and sh, c and ch, t and th, for the words with the letter zh follow only after those with letter z have all been dealt with.

The reduction of the space to 0 has the advantage that the stem obtained by cutting down the suffix is always shorter than the inflected forms of the word.

## 2/ Search for a word, establishment of the word stem

By searching for a word, we use, for the time being, the so-called method of alternatives [3], as no satisfactory method for concentrating [4], [5] and searching for words is available [6].

Let us suppose that words, more precisely word stems are located in the dictionary according to their order of magnitude.

As a first step of searching for a word we compare the first part /first 6 letters/ of the word searched for with the first part of the middle word of the dictionary. /As basis for halving serves the number of information cells, i.e. the number of words, and not the number of cells occupied by the words./ Therefore, if the number of the words is  $N$ , then in the first step the comparison takes place with the first part of the  $\frac{N}{2}$ -th word. In case of full agreement we examine whether the word does continue in the next cell. If not the search for the word has come to an end, if yes, we continue the comparison with the second part of the word etc. If not even the first part of the word did agree, for instance it is smaller, than the middle word of the dictionary, then we compare the examined word with the  $\frac{N}{2} - \frac{N}{4}$ -th word in the first quarter of the dictionary, if it is larger, with the  $\frac{N}{2} + \frac{N}{4}$ -th word in the third quarter of the dictionary. Depending on the result of the comparison, we continue to search for the word upwards or downwards from the place of comparison. We carry on this operation of halving the step space expressing the measure of modification until we attain the machine 0.

As the dictionary contains not full words, only word stems, in case of examination of inflected words we cannot find full, but only partial agreement. In this case we continue the search by separating a letter from the end of the examined word, beginning with the preceding place, but again enlarging the step space to a certain extent. We must reiterate this process until we attain the word stem and find full agreement with the  $j$ -th word of the dictionary. /The process may be accelerated if we consider after the first search how many letters were covered by the partial agreement./ If the word cannot be found in the dictionary /this may be detected for instance if we resected more letters than might figure in the suffix, in the case of verbs in suffix and affix/, then we let write out the word in unchanged form by the machine.

With the aid of this process of searching for words we may establish simultaneously with the search of the word its stem, too, and get the following useful informations:

a/ the Hungarian meaning of the word;

b/ the informations assigned to the verb stem, stored as usual in dictionaries,

c/ we may establish what letters figure in the suffix /for the sake of brevity we mean by suffix the full ending which follows the word stem, nor do we use the concept of word stem in a strictly grammatical sense/;

d/ we may find out the number of letters figuring in the suffix.

### 3/ Contents of the information cell. Classification of words.

We reserve one cell /30 bits/ for every stem of the informations assigned to the word stem. On basis of these cells we are searching for words so they must contain reference where we can find the Russian word and its Hungarian meaning /12 bits/. Besides, the cell must contain informations concerning the "class" and "sub-class" of the word-type /5 bits/, the type of declension /5 bits/ and of connections, governments within the sentence /8 bits/. Informations may indicated, too, by letters for the sake of unity.

By the first letter of the information cell we indicate the following classes and subclasses:

#### I. class /adjectives/

a 00001 adjective

b 00010 relative pronoun

v 00011 possessive pronoun



g	00100	demonstrative or generalizing pronoun
d	00101	personal pronoun
e	00110	preposition
z	00111	numeral

## II. class /nouns/

i	01000	masculine noun denoting an inanimate thing
k	01001	" " " a living thing
l	01010	feminine noun, ending on -a / -я/ denoting an inanimate thing
m	01011	feminine noun, ending on -a / -я/ denoting a living thing
n	01100	feminine noun with soft intonation ending, denoting an inanimate thing
o	01101	feminine noun with soft intonating ending, denoting a living thing
p	01110	neuter noun
r	01111	noun used only in plural

## III. class /verb/

s	10000	verb without variation of stem, I. conjugation
t	10001	" " " " " II. "
u	10010	unchanged stem of verb, I. conjugation
f	10011	changed " " " I. "
x	10100	unchanged " " " II. "
c	10101	changed " " " II. "
y	10110	unchanged stem of verb with r, k stem
j	10111	changed " " " " " "

IV. class

w	11000	modifying word, interjection
q	11001	conjunction
h	11010	adverb
	11011	} others
	11100	
	11101	
	11110	
	11111	

As 2. letter of the information we use only such letter which does not figure in the suffix of the respective class; these denote the particular types of conjugation /the types of government with prepositions/.

In the I. class /adjectives/

b	preposition governing only genitive				
v	"	"	"	prepositive	
d	"	"	"	dative	
z	"	"	"	instrumental	
k	"	"	"	accusative	
l	"	"		accusative or instrumental	
n	"	I.	"	"	prepositive
p	"	governing accusative, dative or prepositive			
r	"	II.	"	"	or prepositive
s	"	governing genitive or instrumental pronominal forms which cannot be treated together with the adjective I.			
c	the same II.				
h	the same III.				

In the II. class /nouns/

- b "стол", "приятель" or "враг" type  
g "поезд" or "доктор" type  
d "пол" type  
z "год" type  
k "лес" type  
l "солдат", "машина" or "яблоко" type  
n "тетрадь" or "путь" type  
p "армия" type  
r "пролетарий" or "комментарий" type  
s "здание" type  
t "слово" type  
f "поле" or "платье" type  
e "облако" type  
h "время" type

In the III. class /verbs/

- b "читать", "уметь" or "нести" type  
d "стучать", "говорить" or "гнать" type  
z "стоять" type  
k "рисовать", "писать", "стать", "крыть" or  
"начать" type  
p "вернуть" type  
r "помочь" or "протечь" type  
f reserve  
c "

Here we must remark that this is only a provisional arrangement of types, prepared for the experimental model. In some group types of declension of very different kinds were put together /for instance: солдат, машина, яблоко/ on basis of some coincidence in declension /nominative plural солдаты, машины, яблоки genitive plural солдат, машин, яблок/ yet this does not cause any trouble in the course of analysis, because they can easily be separated on basis of the first letter of the information. /Masculine noun denoting a living thing, feminine noun ending with a denoting an inanimate thing, neuter noun belong all to other subclasses./

The selection of types and sub-classes for distinction depends on what categories we assign to the single bits. Our purpose is to utilize optimally the bits at our disposal, to create as much categories as possible on basis of available information. We strove for the separation of categories, of course, only if it had any importance with respect to the translation into Hungarian, as our main purpose was not grammatical analysis.

We have assigned the following categories to the 30 bits at our disposal:

## II. class /nouns/

- bit 1. nominative singular masculine /N/
- " 2. accusative " " inanimate /A/
- " 3. genitive singular masculine or neuter /G/
- " 4. dative " " " /D/
- " 5. instrumental " " " /I/
- " 6. prepositive singular masculine ending with -y / -Ю /  
/P/
- " 7. nominative singular feminine ending with -a / -Я /

- bit 8. accusative singular feminine ending with -y / 0 /
- " 9. genitive " " "
- " 10. dative " " " " -e
- " 11. instrumental " " "
- " 12. " " masculine, feminine or neuter  
ending with -e
- " 13. nominative plural ending with -H / H /
- " 14. accusative " " " " "
- " 15. genitive plural without suffix
- " 16. dative "
- " 17. instrumental "
- " 18. prepositive "
- " 19. nominative singular neuter
- " 20. accusative " "
- " 21. nominative " feminine ending with -h
- " 22. accusative " " " " -h
- " 23. dative singular ending with -H
- " 24. prepositive " " " -H
- " 25. nominative plural " " -a / a /
- " 26. accusative " " " -a "
- " 27. genitive " with suffix
- " 28. accusative " " " living
- " 29. " " without " "
- " 30. " singular masculine; living

So we had to reserve 9 bits for the accusative /singular-plural, living-inanimate, masculine-feminine-neuter, with and

without suffix/, while, on the other hand, instrumental has only 3 bits. This reflects in no way the importance of the particular cases, but may be explained by considerations of the possibility and necessity of distinction. The instrumental singular of words with -ий stem of different gender /коммерци-ский, армия, здание/ is drawn together, on the other hand we had to segregate for instance the mode of formation with and without suffix of genitive plural.

With regard to the I. class /adjectives/ the same assignment holds true in order to enable us to continue directly the analysis after the morphologic analysis of each single word by the collation of the noun with the adjective belonging to it.

With regard to the III. class /verbs/

- |        |                                    |       |              |                            |
|--------|------------------------------------|-------|--------------|----------------------------|
| bit 1. | present /simple future/            | tense | first person | singular                   |
| " 2.   | "                                  | "     | second       | "                          |
| " 3.   | "                                  | "     | third        | "                          |
|        | /only first conjugation/           |       |              |                            |
| " 4.   | present /simple future/            | tense | first person | plural                     |
| " 5.   | "                                  | "     | second       | first conjugation          |
| " 6.   | "                                  | "     | third        | "                          |
| " 7.   | "                                  | "     | second       | plural, second conjugation |
| " 8.   | "                                  | "     | second       | plural, "                  |
| " 9.   | past tense, masculine              |       |              |                            |
| " 10.  | " " feminine                       |       |              |                            |
| " 11.  | " " neuter                         |       |              |                            |
| " 12.  | " " plural                         |       |              |                            |
| " 13.  | imperative, second person singular |       |              |                            |

- bit 14. imperative, second person plural with ending -ите
- " 15. " " " " " " -йте,ьте
- " 16. infinitive with ending -ТЬ
- " 17. " " " " -ТИ
- " 18. " " vowel+ending ТЬ /with the exception  
of -ОТЬ /
- " 19. " with ending еТЬ
- " 20. participle present tense, active
- " 21. " past " "
- " 22. " present " passive
- " 23. " past " " full form with affix  
- еНН-, -НН-
- " 24. " " " " full form with affix  
- Т-
- " 25. " " " " shortened form,  
masculine
- " 26. " " " " shortened form,  
feminine
- " 27. " " " " shortened form,  
neuter
- " 28. " " " " shortened form,  
plural
- " 29. imperfective adverb
- " 30. perfective adverb

In denominating the categories we disregarded, for brevity's sake, grammatical precision and denominations indicate only differences between the categories.

#### IV. EXPOSITION OF THE MODEL OF MACHINE ANALYSIS II.

##### /Operation of the model/

As a result of the process of searching for words we attained the contents of the information cell and the letters figuring in the ending are available.

Our first task is to examine on basis of the first two bits of the information cell to what class examined word belongs.

If the first two bits

00 - I. class /adjective/

01 - II. " /nouns/

10 - III. " /verbs/

11 - IV. " /others/.

Words belonging to the first 3 classes are examined in an identical way, by the method described in the II. part, but with different Boole-vectors in any of the three classes. These vectors are assigned to letters in the endings. The Boole-vector assigned to letter m of the substantive ending contains for instance 1 in the place of those categories in the ending of which there is m:

singular	singular	plural	singular	p p p p p s
N A G D I P	N A G D I P	N A G D I P	N A N A D P	N A G A A A
0 0 0 0 1 0	0 0 0 0 0 0	0 0 0 1 1 0	0 0 0 0 0 0	0 0 0 0 0 0

As an example, let us examine how the word числом is analysed. At the stem chisl we find the informations concerning the word. We discover from the first two bits of the information cell that the word is a noun. This means the point of departure to find the corresponding Boole-vectors. /This point



of departure shall be for the I. class /adjectives/ cell  $\alpha$ , for the II. class /nouns/ cell  $\beta$ , for the III. class /verbs/ cell  $\gamma$ . The Boole-vector belonging to letter  $\underline{o}$  of the ending ом of числом will be found so that we use for the modification of address  $\beta$  the letter  $\underline{o}$  itself as a number /01101/: the  $\beta + \underline{o}$  -th cell contains the Boole-vector belonging to the letter  $\underline{o}$  of the noun ending:

	sing.	sing.	plur.	sing.	ppppps
	NAGDIP	NAGDIP	NAGDIP	NANADP	NAGAAA

$$/\beta + \underline{o}/ = 000010 \ 000000 \ 000000 \ 110000 \ 000000$$

Similarly we found in the  $\beta + \underline{m}$  -th cell the Boole-vector belonging to letter  $\underline{m}$  of the noun ending

	sing.	sing.	plur.	sing.	ppppps
	NAGDIP	NAGDIP	NAGDIP	NANADP	NAGAAA

$$/\beta + \underline{m}/ = 000010 \ 000000 \ 000110 \ 000000 \ 000000$$

Performing logic multiplication with the contents of these two cells we find 1 on the place of those categories in the ending of which both letters  $\underline{o}$  and  $\underline{m}$  occur:

	sing.	sing.	plur.	sing.	ppppps
	NAGDIP	NAGDIP	NAGDIP	NANADP	NAGAAA

$$/\beta + \underline{o} / \wedge / \beta + \underline{m} / = 000010 \ 000000 \ 000000 \ 000000 \ 000000$$

In the given case there is only such a category instrumental singular masculine-neuter.

The applied method led therefore here already after the first logic multiplication to an unambiguous result, the later logic multiplications do no more modify this result.

But the performance of further logic is not superfluous for instance in the case of the word числам:

	sing.	sing.	plur.	sing.	ppppps
	NAGDIP	NAGDIP	NAGDIP	NANADP	NAGAAA
$\beta + \underline{a}$ =	001000	100000	000111	000000	110001
$\beta + \underline{m}$ =	000010	000000	000110	000000	000000
$\beta + \underline{a} \wedge \beta + \underline{m}$ =	000000	000000	000110	000000	000000

The result is not unambiguous, for both in the cases of dative plural and instrumental plural the letters a and m occur alike /числам, числами/.

Fixing the number of letters in the ending /now: 2/ we define a new Boole-vector:

	sing.	sing.	plur.	sing.	ppppps
	NAGDIP	NAGDIP	NAGDIP	NANADP	NAGAAA
$\beta - 2$ =	000010	000010	000101	000000	001100

This vector contains 1 on the place of those categories whose ending consists of 2 letters. Performing a new logic multiplication with this vector:

	sing.	sing.	plur.	sing.	ppppps
	NAGDIP	NAGDIP	NAGDIP	NANADP	NAGAAA
$\beta + \underline{a} \wedge \beta + \underline{m} \wedge \beta - 2$ =	000000	000000	000100	000000	000000

In the given case the second logic multiplication led to the unambiguous result, and further logic multiplications modify no more the same.

Yet the analysis of, for instance, the word предмета shows that in general further logic multiplications are required, too.

	sing.	sing.	plur.	sing.	ppppps
	NAGDIP	NAGDIP	NAGDIP	NANADP	NAGAAA
$/\beta + \underline{a}/ =$	001000	100000	000100	000000	110001
$/\beta - \underline{1}/ =$	111101	111101	111000	111111	110011
$/\beta + \underline{a}/ \wedge / \beta - \underline{1}/ =$	001000	100000	000000	000000	110001

The result of logic multiplication shows that substantive ending containing letter a occurs in genitive singular masculine-neuter, in nominative singular ending with feminine a, in nominative and accusative plural of neuter and certain masculine nouns, as well as in accusative singular of masculine nouns denoting living things.

But we may continue the analysis by using the first two letters /sub-class and type/ of the information cell.

Let us consider first the type, that is the 2nd letter of the information cell. We use only such letters to indicate the type which do not occur in the ending of the respective class. We are using these letters the same way to define the Boole-vector belonging to the type than the letters in the ending. The word числам belongs to the type b, therefore

	sing.	sing.	plur.	sing.	ppppps
	NAGDIP	NAGDIP	NAGDIP	NANADP	NAGAAA
$/\beta + \underline{b}/ =$	111110	000001	110111	000000	001101

Performing the corresponding logic multiplication:

	sing.	sing.	plur.	sing.	ppppps
	NAGDIP	NAGDIP	NAGDIP	NANADP	NAGAAA

$$\beta_{+a} / \beta_{-1} / \beta_{+b} / = 001000 \ 000000 \ 000000 \ 000000 \ 000001$$

The unambiguousness is still not secured; the substantive ending with one letter containing letter a of substantives belonging to this type may indicate genitive singular masculine-neuter or accusative singular of masculine living things.

But let us consider the first letter of the information, too. Concerning the word предмет it is i. In the i-th cell calculated from the k cell of departure common to the three classes we find the following Boole-vector:

	sing.	sing.	plur.	sing.	ppppps
	NAGDIP	NAGDIP	NAGDIP	NANADP	NAGAAA

$$/k + i/ = 111111 \ 000001 \ 111111 \ 000011 \ 111000$$

covering masculine substantives which denote inanimate things.

The last logic multiplication has eventually the following result:

$$\beta_{+a} / \beta_{-1} / \beta_{+b} / k + i / 001000 \ 000000 \ 000000 \ 000000 \ 000000$$

From the obtained vector it may be established that the word предмета stands in genitive singular.

Logic multiplication had therefore to be performed with the following Boole-vectors:

- 1/ the vectors of the letters belonging to the ending
- 2/ the vector assigned to the number of the letters in the ending

3/ with the vector assigned to the type, and finally

4/ " " " " " " sub-class

The result will not always be unambiguous, of course. For instance referring to the word числа /letter in the ending: a, number of the letters in the ending: 1, type: t, sub-class: p:

	sing.	sing.	plur.	sing.	ppppps
	NAGDIP	NAGDIP	NAGDIP	NANADP	NAGAAA
/β + a/ =	001000	100000	000111	000000	110001
/β - 1/ =	111101	111101	111000	111111	110011
/β + t/ =	001110	000001	001111	110000	110000
/κ + p/ =	001110	000001	111111	110011	111000

Their logic multiplication:

числа → 001000 000000 000000 000000 110000

The form числа can be therefore - according to the analysis - genitive singular masculine-neuter, nominative plural neuter or accusative plural neuter.

But if this word is a part of a nominal syntagma in which an adjective or a pronoun also refers to the noun, in agreement with the same, or is preceded by a preposition, than we may continue the analysis on a higher level.

Let us suppose, for instance, that the word числа is preceded by the adjective целые. With respect to this word:

	sing.	sing.	plur.	sing.	ppppps
	NAGDIP	NAGDIP	NAGDIP	NANADP	NAGAAA

/d + y/ = 110010 000000 111111 000000 111110

/d + e/ = 001101 001111 110010 110011 110001

/d + y/ ^ /d + e/ = 000000 000000 110010 000000 110000

/d - a/ = 110011 111111 111101 111111 111110

In last analysis we obtain therefore the following vector concerning the word числа:

/d + y/ ^ /d + e/ ^ /d - a/ = 000000 000000 110000 000000 110000

Conferring this with the vector obtained for the word

числа → /001000 000000 000000 000000 110000/

we obtain the following result:

	sing.	sing.	plur.	sing.	ppppps
	NAGDIP	NAGDIP	NAGDIP	NANADP	NAGAAA

~~целые числа~~ → 000000 000000 000000 000000 110000

The expression in question may therefore stand in nominative or accusative plural. More about it we can discover without analysing the structure of the sentence if, in addition, a preposition belongs to this nominal syntagma, e.g. через целые числа

To the preposition через belongs the following vector:

	sing.	sing.	plur.	sing.	ppppps
	NAGDIP	NAGDIP	NAGDIP	NANADP	NAGAAA

через           → 010000 010000 010000 010100 010111

Performing the corresponding logic multiplication:

через целые числа   → 000000 000000 000000 000000 010000

The expression has grammatically but one meaning.

In conclusion let us consider a verbal example.

Let us examine a participle, for this will be more interesting. For instance the word to be analysed should be изменяющего. In the dictionary we find only the form изменя. Performing the logic multiplication with the vectors of the letters belonging to the ending resp. affix

/y + w/ = 100001 000000 000000 011111 000000

/y + t/ = 001011 110000 011101 110001 000000

/y + h/ = 010000 000000 000000 011000 000001

/y + e/ = 011110 011111 011000 111110 111101

/y + g/ = 000000 000000 000000 011111 000000

/y + o/ = 000000 000010 000001 011111 001000

are the vectors in question, their logic multiplication is

изменяющего           → 000000 000000 000000 010000 000000

Bit 20. indicates, under our agreement, the present participle active of verbs. But as we are dealing with a participle, we may carry on the analysis on adjective level, we must only segregate the letters belonging to the affix of the participle and need to use the vectors of the letters belonging to the ending for further analysis:

	sing.	sing.	plur.	sing.	ppppps
	NAGDIP	NAGDIP	NAGDIP	NANADP	NAGAAA

/d + e/	=	001101	001111	110010	110011	110001
---------	---	--------	--------	--------	--------	--------

/d + g/	=	001000	000000	000000	000000	000001
---------	---	--------	--------	--------	--------	--------

/d + o/	=	111101	001111	000000	110011	000001
---------	---	--------	--------	--------	--------	--------

are the corresponding vectors, their logic multiplication is

	sing.	sing.	plur.	sing.	ppppps
	NAGDIP	NAGDIP	NAGDIP	NANADP	NAGAAA

<u>изменяющего</u>	→	001000	000000	000000	000000	000001
--------------------	---	--------	--------	--------	--------	--------

The analysis therefore established that the form stands in genitive singular masculine-neuter or accusative singular masculine-neuter denoting living things.

--

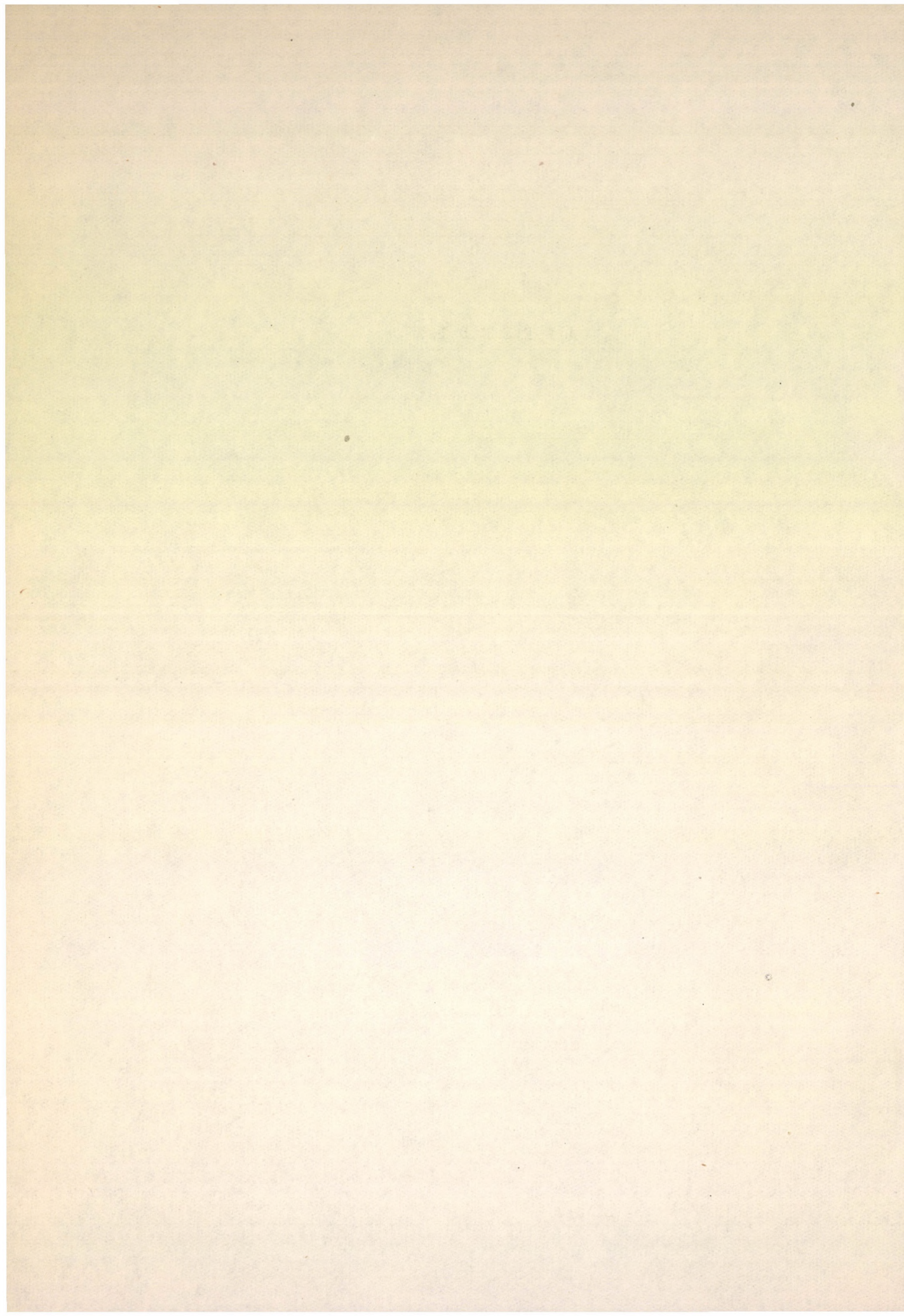


## References

- [1] In respect of the dissolution of grammatical ambiguity by help of logical multiplication see the paper of I.I.Revzin  
И.И.РЕВЗИН: Структуральная лингвистика, семантика и проблемы изучения слова,  
Вопросы языкознания, VI./1957/ 2. стр.1291-1294.
- [2] In respect of the use of logical multiplication for the identification of strings of symbols see the paper of B.Dömölki /Jelsorozatok tulajdonságainak felismerésére szolgáló algoritmusok, Az MTA Számítástechnikai Központjának Tájékoztatója, 8.sz. 1962, 63-88 o. in Hungarian/ and the same summarized by F.Kiefer in German in the Appendix of this book.
- [3] See: Г.Г.СТЕЦЮРА: Новый принцип построения запоминающих устройств, ДАН, 1960. т. 131. № 6.  
стр. 1292-1294.
- [4] Cf. W.S.Cooper: The Storage Problem, Mechanical Translation, 1953. and
- [5] Л.Н.КОРОЛЕВ: О переключающей функции устройства поисков по таблице, ДАН, 1958. т. 126. № 8.
- [6] About the problem of dictionary-look-up a survey is to be found in the paper Л.А.КАЛУЖНИН, А.А.СТОГНИЙ - Л.С.СТОЙКОВ: О принципах построения машинных словарей, Прикладная лингвистика и машинный перевод, Киев, 1962, стр. 6-20.



A P P E N D I X



ALGORITHMUS FÜR DIE IDENTIFIZIERUNG VON  
SYMBOLFOLGEN

(Auf Grund B.Dömölki's Aufsätze zusammengefasst von F. Kiefer)

Grundbegriffe

Wir bezeichnen die Menge der Symbole

$$A = a_1, a_2, \dots, a_n$$

als Alphabet. Eine Folge von Symbolen

$$S = a_{s_1} a_{s_2} \dots a_{s_q} \quad \text{wobei} \quad 1 \leq s_i \leq n$$

$$\text{und} \quad 1 \leq i \leq q$$

sind, wird als ein Wort der Länge  $q$  bezeichnet.

Wir bezeichnen die Menge aller Wörter des Alphabets  $A$  mit  $S/A$ .

Es sei

$$T = \left\{ \begin{array}{cccc} t_{11} & t_{12} & \dots & t_{1p} \\ t_{21} & t_{22} & \dots & t_{2p} \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ t_{h_{11}} & t_{h_{22}} & & t_{h_{pp}} \end{array} \right\}, \text{ kurz } T = \{ t_{ij} \}$$

eine Tabelle aus den Symbolen des Alphabets  $A$ , die  $p$  Spalten und in jeder Spalte  $h_j$  Elemente hat. Zum Beispiel

$$\left( \begin{array}{cccc} a_1 & a_1 & a_3 & a_6 \\ a_3 & a_3 & & a_7 \\ & a_5 & & a_1 \\ & & & a_2 \end{array} \right),$$

Die Zahl der Spalten - wir bezeichnen sie mit  $p$  - heisst die Länge der Tabelle. Die einzelnen Spalten werden mit den Symbolen

$$t_1, t_2, \dots, t_p$$

bezeichnet. Jede Spalte ist die Vereinigung gewisser Symbole, deren Zahl  $h_j$  beträgt.

$$t_j = \bigcup_{k=1}^{h_j} \{t_{jk}\} \quad \text{für } 1 \leq j \leq p$$

Wir sagen von einem

$$X = X_1 X_2 \dots X_p \in S/A/$$

Wort, dass es in der Tabelle  $T$  vorkommt, wenn

$$x_j \in t_j \quad \text{für } 1 \leq j \leq p$$

gilt. Die Menge aller solcher Wörter bezeichnen wir mit  $\tilde{T}$ .

Es sei zum Beispiel

$$A = \{a, b, c, d\}$$

(jetzt ist  $a_1 = a$ ,  $a_2 = b$ ,  $a_3 = c$  und  $a_4 = d$ ) und

$$T = \left( \begin{array}{ccc} b & a & a \\ c & d & b \\ & & d \end{array} \right)$$

Für die  $h_j$  Werte gilt nun

$$h_1 = 2, h_2 = 2, h_3 = 3$$

und für die einzelnen Spalten

$$b \in t_1, \quad c \in t_1, \quad a \in t_2, \quad d \in t_2,$$

$$a \in t_3, \quad b \in t_3, \quad d \in t_3$$

In der Tabelle T kommen nun zum Beispiel folgende Wörter vor:

$$baa, cdb, bad$$

aber die Wörter

$$\underline{aaa}, \underline{bbd}, \underline{cac}$$

kommen in ihr nicht vor, da die unterstrichenen Symbole in den entsprechenden Spalten nicht vorkommen, d.h.

$$a \notin t_1, \quad b \notin t_2, \quad c \notin t_3$$

Bilden wir die aus  $n$  Reihen und  $p$  Spalten bestehende Boolesche Matrix

$$B = \left\{ b_{ij} \right\}, \quad \text{wobei } 1 \leq i \leq n \quad \text{und} \\ 1 \leq j \leq p \quad \text{ist, (kurz: B-Matrix)}$$

folgendermassen:

$$b_{ij} = \begin{cases} 1, & \text{wenn } a_i \in t_j \\ 0 & \text{wenn } a_i \notin t_j \end{cases}$$

Die Matrix B unseres Beispiels ist nun

$$B = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

Betrachten wir die einzelnen Reihen

$$B_1, B_2, \dots, B_n$$

der Matrix B als Boolesche Vektoren von p Elementen und erklären wir unter diesen die Konjunktion, Disjunktion und Negation in üblicher Weise. Das heisst, wenn

$$X = (x_1, x_2, \dots, x_p)$$

und

$$Y = (y_1, y_2, \dots, y_p)$$

zwei Boolesche Vektoren sind, dann gelte

$$X \wedge Y = (x_1 \wedge y_1, x_2 \wedge y_2, \dots, x_p \wedge y_p)$$

für die Konjunktion,

$$X \vee Y = (x_1 \vee y_1, x_2 \vee y_2, \dots, x_p \vee y_p)$$

für die Disjunktion, und

$$\bar{X} = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_p)$$

für die Negation.

Es seien

$$E_1 = (1, 0, \dots, 0)$$

$$E_2 = (0, 1, \dots, 0)$$

.....

$$E_p = (0, 0, \dots, 1)$$

die entsprechenden Einheitsvektoren und bedeute

$$\underline{0} = (0, 0, \dots, 0)$$

den Nullvektor.

Ein

$$Y \in S/A/$$



Wort ist von Typ T, wenn

$$Y = EXF$$

ist, und

$$X \in \tilde{T}$$

(E, F sind beliebige Wörter, können daher auch leer sein.  
Und E, F, X  $\in$  S/A/.)

So sind zum Beispiel in Bezug auf die obige Tabelle T  
die Wörter

acbadac, bbcdab, bdd

von Typ T, denn die unterstrichenen Teile der Wörter (X)  
kommen in der Tabelle T vor.

### Satz 1.

Es gilt folgender Satz:

Ein Wort S ist dann und nur dann von Typ T, wenn

$$Z_q \wedge E_p \neq 0,$$

wobei die Vektoren  $Z_k$  mit folgender Rekursion gebildet werden

$$Z_0 = 0$$

$$Z_k = (Z_{k-1} \cdot 2^{-1} \quad E_1) \wedge B_{s_k}$$

Die Multiplikation mit  $2^{-1}$  bedeutet eine Rechtsverschiebung, und  $B_{s_k}$  ist der entsprechende Boolesche Vektor.

Nehmen wir zum Beispiel das Wort

acbad

und bilden die  $x_c$  Vektoren  $Z_k$

$$z_0 = (0, 0, 0)$$

$$z_1 = (1, 0, 0) \wedge (0, 1, 1) = (0, 0, 0)$$

$$z_2 = (1, 0, 0) \wedge (1, 0, 0) = (1, 0, 0)$$

$$z_3 = (1, 1, 0) \wedge (1, 0, 1) = (1, 0, 0)$$

$$z_4 = (1, 1, 0) \wedge (0, 1, 1) = (0, 1, 0)$$

$$z_5 = (1, 0, 1) \wedge (0, 1, 1) = (0, 0, 1)$$

und

$$z_5 \wedge e_3 = (0, 0, 1) \neq \underline{0}$$

- . -

Wir betrachten nun den Fall, wenn die Erfüllung mehrerer Eigenschaften ähnlicher Natur untersucht werden sollte.

Es seien

$$T^1, T^2, \dots, T^m$$

die entsprechenden Tabellen, deren Länge der Reihe nach

$$p_1, p_2, \dots, p_m$$

beträgt, und die zu diesen gehörenden B-Matrixen der Reihe nach

$$B_1, B_2, \dots, B_m$$

Bilden wir die zusammengesetzte B-Matrix durch das Nebeneinanderreihen der einzelnen  $B_i$ -Matrixen. Auf diese Weise erhalten wir eine Matrix, die  $n$  Reihen und

$$p = p_1 + p_2 + \dots + p_m$$

Spalten hat.

Bilden wir ausserdem die Vektoren  $K$  und  $V$  folgendermassen:

a) Der Vektor  $K$  enthalte in den Stellen

$$1, p_1 + 1, p_1 + p_2 + 1, \dots, p_1 + p_2 + \dots + p_{m-1} + 1,$$

die Eins (d.h. in den Stellen, die den ersten Spalten der einzelnen Tabellen  $T^i$  entsprechen, also dort, wo die einzelnen Tabellen  $T^i$  beginnen).

b) Der Vektor  $V$  enthalte in den Stellen

$$p_1, p_1 + p_2, p_1 + p_2 + p_3, \dots, p_1 + p_2 + \dots + p_m$$

die Eins (d.h. in den Stellen, die den letzten Spalten der einzelnen Tabellen  $T^i$  entsprechen, also dort, wo die einzelnen Tabellen  $T^i$  enden).

Bei Vektoren haben  $p$  Elemente.

Es seien zum Beispiel

$$T^1 = \begin{pmatrix} b & a & a \\ c & d & b \\ & & d \end{pmatrix}$$

$$T^2 = \begin{pmatrix} a & b \\ & d \end{pmatrix}$$

$$T^3 = (c)$$

$$T^4 = \begin{pmatrix} & a & & \\ a & b & c & \\ & & d & \\ & c & & \\ & d & & \end{pmatrix}$$

(Jetzt ist  $a_1 = a$ ,  $a_2 = b$ ,  $a_3 = c$ ,  $a_4 = d$  und  $p_1 = 3$ ,  $p_2 = 2$ ,  $p_3 = 1$ ,  $p_4 = 3$ .) Die entsprechenden B-Matrizen sind

$$B_1 = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix} \quad B_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \quad B_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \quad B_4 = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

die zusammengesetzte Matrix

$$P = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}$$

die Vektoren K und V sind nun

$$K = (1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0) \\ V = (0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1)$$

### Satz 2.

Bilden wir die  $Q_k$  Vektoren mit folgender Rekursion

$$Q_0 = 0$$

$$Q_k = \left\{ \left[ (Q_{k-1} \wedge \bar{V}) 2^{-1} \right] \vee K \right\} \wedge B_{sk}$$

Es gilt folgender Satz:

Das Wort

$$S = a_{s_1} a_{s_2} \dots a_{s_q}$$

ist dann und nur dann von Typ  $T^r$ , wobei  $r$  eine feste Zahl zwischen 1 und  $m$  ist, wenn

$$Q_q \wedge V \neq \underline{0}$$

und das Wort  $S$  ist gerade von Typ  $T^r$ , wenn das

$$p_1 + p_2 + \dots + p_r =$$

te Element des Vektors  $Q_q$  eben Eins ist.

Betrachten wir zum Beispiel das Wort

aabeda

und die Matrizen  $T^1$ ,  $T^2$ ,  $T^3$  und  $T^4$  des vorigen Beispiels. Dann ist

$$Q_0 = (000 \ 000 \ 000)$$

$$Q_1 = (000 \ 100 \ 100) \quad \text{und} \quad Q_1 \wedge V = (000 \ 000 \ 000)$$

$$Q_2 = (000 \ 100 \ 110) \quad \text{und} \quad Q_2 \wedge V = (000 \ 000 \ 000)$$

$$Q_3 = (100 \ 010 \ 010) \quad \text{und} \quad Q_3 \wedge V = (000 \ 010 \ 000)$$

Da nun das

$$p_1 + p_2 = 3 + 2 = 5$$

te Element Eins ist, ist das Wort aab von Typ  $T^2$ .

$$Q_4 = (100 \ 001 \ 001) \quad \text{und} \quad Q_4 \wedge V = (000 \ 001 \ 001)$$

Da nun das

$$p_1 + p_2 + p_3 = 3 + 2 + 1 = 6$$

te und auch das

$$p_1 + p_2 + p_3 + p_4 = 3 + 2 + 1 + 3 = 9$$

te Element Eins ist, so ist das Wort aabc sowohl von Typ  $T^3$  als auch von Typ  $T^4$ .

Das Verfahren weiterführend erhält man

$$Q_5 = (010 \ 000 \ 000) \quad \text{und} \quad Q_5 \wedge V = (000 \ 000 \ 000)$$

$$Q_6 = (001 \ 100 \ 100) \quad \text{und} \quad Q_6 \wedge V = (001 \ 000 \ 000)$$

Dies bedeutet, dass das Wort

aabeda.

von Typ  $T^1$  ist, da jetzt das  $p_1 = 3$ -te Element Eins wurde.

C O N T E N T S

Preface . . . . .	3
G. Hell:	
Recognition of Nominal Groups in MT from Russian into Hungarian. . . . .	5
G. Sipőczy:	
Algorithms for Analysis of Homonym Nominal Forms . . . . .	109
F. Kiefer:	
Algorithm and Program for Finding a Word in the Machine Vocabulary . . . . .	189
F. Kiefer:	
Algorithm for Analysis of Endings. . . . .	203
D. Varga:	
Morphological Analysis by Help of the Method of Successive Delimitation . . . . .	223
APPENDIX	
B. Dömölki - F. Kiefer:	
Algorithms for the Identifying of Symbol Sequences . . . . .	259



1315  
Abelstein