

tanulmányok **97/1979**

MTA Számítástechnikai és Automatizálási Kutató Intézet Budapest



MAGYAR TUDOMÁNYOS AKADÉMIA
SZÁMITÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZETE

BOOLE-FÜGGVÉNY KEZELŐ RENDSZER

Irta:

MATAVOVSZKY TIBOR
dr PÁSZTORNÉ VARGA KATALIN

Tanulmányok 97/1979

A kiadásért felel:

DR VÁMOS TIBOR

ISBN 963 311 089 0

ISSN 0324-2951

Készült a SZÁMOK
Reprográfiai Üzemében 9282

Tartalomjegyzék

Boole-függvény kezelő rendszer

	old
Bevezetés	5
1. A Boole-függvény kezelő rendszer általános leírása	7
2. A programrendszer általános leírása	15
2.1 A függvénykezelő rendszer adatstruktúrája	17
2.2 Adatstruktúra kialakító műveletek	35
2.3 Függvények feletti műveletek	38
3. A realizált programrendszer	42
3.1 A műveletszervezést és a struktúraátalakítást biztosító programok	42
3.2 A BOOLOP rutin által aktivizált, függvények feletti műveletek	99
3.2.1 HERT művelet	99
3.2.2 CHELY művelet	114
3.2.3 EGYSZ művelet	123
3.2.4 MIN művelet	123
3.2.5 BDIFF művelet	123
3.2.6 XOR művelet	124
4. A Boole függvény kezelő rendszer aktivizálása	125
5. Példák a BFKR működésére	135
Irodalomjegyzék	148

BOOLE-FÜGGVÉNY KEZELŐ RENDSZER

BEVEZETÉS

A Boole-függvények vizsgálata, osztályozása, e függvények feletti műveletvégzésre és azonos átalakítások elvégzésére hatékony módszerek keresése fontos témakör. Ilyen problémák megoldásával találkozunk minden olyan tudományos és technikai feladat kapcsán, ahol a feladat vagy annak egy része leírható a matematikai logika formalizmusával. Ilyen feladatokat vet fel a diszkrét digitális berendezések tervezése, analizise, a keresési eljárások formalizálása, a formális tételbizonyítás, fizikai rendszerek működési hibáinak keresése, stb.

Ennek megfelelően az egyik nagy feladatcsoportot az automataelmélet alkalmazásai adják. Az automataelméletben a diszkrét digitális berendezéseket egyszerű /építő/ elemekből álló rendszernek tekintjük.

Alapproblémák:

- Logikai szintézis, amelynek az a célja, hogy a megadott funkciót teljesítő strukturát előállítsa. E feladat során

az adott függvény dekompozícióját keressük az építőelemekre vonatkozóan. Ez lényegében Boole-függvényrendszer kezelést jelent, amelynek eredményeképpen az építőelemeket mint műveleti jeleket tartalmazó formulát kapunk. A formulára lehetséges bizonyos korlátozások feltevése is.

- Hibadetektálás diszkrét rendszerekben a Boole-függvényrendszerek feletti operációk végzését eszközként használja detektáló bemenőjelek előállítására.

A Digitális Osztályon kidolgozott Boole-függvény kezelő rendszer célja az említett feladatokhoz eszköz szolgáltatása. A rendszer a realizált vagy adott alakban vizsgálandó függvény struktúrájának számítógépben való megőrzését is biztosítja a lehetőségek határain belül egy célorientált, kapcsolt listastruktúrában való adattárolás segítségével [2].

Hasonló célú függvények feletti operációkat végző rendszer kutatása és realizálása ismert [1]. A probléma megközelítése azonban ott más.

1. A Boole-függvény kezelő rendszer általános leírása

Azokban a feladatokban, amelyeknek megoldása során segédeszköz-ként egy ilyen rendszer szolgáltatásai jól felhasználhatók a függvények megadási módja különböző. Valamint a függvény kezelő rendszeren belül az egyes feladatok megoldása is megkövetelhet speciális függvényábrázolást. Ezért a függvények feletti operációk egy része a különböző leírások közötti konverzió.

A függvény kezelő rendszerben a függvényekre megvalósított műveletek két nagy csoportra oszthatók:

- azonos átalakítások
- függvények közötti műveletek

A függvény kezelő rendszer kidolgozása során első célunk a normálformában megadott Boole-függvényekre egy sor művelet elvégzése. A műveleteket mind teljesen meghatározott, mind parciális / Φ / Boole-függvényekre értelmeztük.

Megjegyzés: Az $f \in \Phi$ -Boole-függvény megadása a szokásos

$$f = f_1 \vee \chi f_\emptyset \quad \text{vagy} \quad f = f_1 \vee \chi \bar{f}_0$$

megadási módoknak megfelelően - ahol f_1, f_0, f_\emptyset rendre az f 1, 0 és határozatlansági pontjainak karakterisztikus

függvényei - az $\{f_1, f_\emptyset\}$ vagy az $\{f_1, f_0\}$ függvénytűpárokkal történhet a rendszerben a végrehajtandó feladattól függően.

E Boole-függvény kalkuluban az egyes változókat a gépi szóban az egyes helyértékek reprezentálják. Az elemi tagokat /elemi konjunkció/ és elemi tényezőket /elemi diszjunkció/ két bitsorozattal reprezentáljuk.

Az első komponens ad információt arról, hogy az általa reprezentált változó negálva /0/ vagy negálatlanul /1/ szerepel.

A második komponens jelzi, hogy az illető helyértékhez tartozó változó szerepel-e /1/ vagy nem /0/. Tehát $\begin{matrix} 0 \\ 1 \end{matrix} - \bar{x}$, $\begin{matrix} 1 \\ 1 \end{matrix} - x$, $\begin{matrix} 0 \\ 0 \end{matrix} -$ nem szerepel x . Pl. az $a\bar{b}c$ vagy $a\bar{b}cde$ elemi kifejezésnek, ha a függvényben az a, b, c, d, e, f változók szerepelnek és a megfeleltetés $a-2^0$, $b-2^1$, $c-2^2$, $d-2^3$, $e-2^4$, $f-2^5$;

I 010001

II 011001

Világos, hogy a több függvény közötti műveletek csak akkor végezhetőek el, ha egyforma változóknak ugyanazon helyérték felel meg.

Megengedjük kitűntetett normálformák helyértékes megadását is. Ebben a megadási módban egy mintermet /maxtermet/ egy szóban reprezentálunk. A megfelelő helyértéken 0 /vagy 1/ áll, ha a

helyérték által reprezentált változó negálva /vagy negálatlanul/ szerepel az illető termben. A nem kitüntetett normálformában /DNF-ben, KNF-ban/ megadott függvény helyértékes leírására kétféle formát vezettünk be. Az első forma szerint egy Boole-függvény megadása lényegében a változó lista, a változók helyértékes megfeleltetése és az elemi kifejezések I és II komponensei listája megadását jelenti. A második forma a normálforma elemi kifejezéseinek az első forma szerinti felsorolását jelenti. A két megadási mód illusztrációját az 1. ábrán láthatjuk.

a - 0	00011	a - 0	00001	00000	00001
b - 1	00011	b - 1	00001	00001	00001
c - 2	01100	c - 2	00010	00100	00000
d - 3	01101	d - 3	00010	00100	01000
e - 4	10001	e - 4		01000	10000
	11001			01000	10000
1. megadási mód		2. megadási mód			

1. ábra

$$f = ab \vee \bar{a}cd \vee a\bar{d}e \text{ helyértékes ábrázolásai}$$

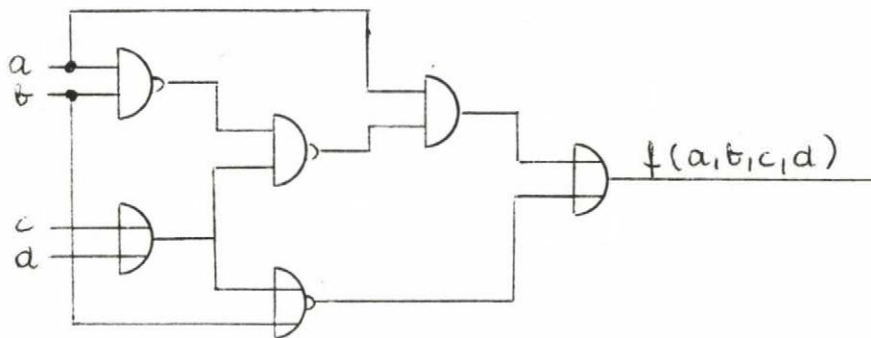
A normálformában adott függvények helyértékes ábrázolása bitsorozatok vagy bitsorozatpárok halmaza, ezért a függvényműveletek bitsorozatok kezelésével oldhatók meg. A függvénykezelő rendszerhez ennek megfelelően bitsorozatokkal végzett műveletek is tartoznak. Ezek közül a fontosabbak:

- a. két n hosszúságu bitsorozatra
 - . bitenkénti konjunkció
 - . bitenkénti diszjunkció
 - . bitenkénti kizáró vagy
- b. egy n hosszúságu bitsorozatra
 - . bitenkénti negáció
 - . k -adik bit $/0 < k \leq n/$ törlése
 - . k -adik bit $/0 < k \leq n/$ kiválasztása

A következő célunk a Boole-függvényeket leíró tetszőleges formulák kezelésének valamint e formulákon és formulákkal való műveletvégzés megvalósítása. Így egy rugalmasabb függvénykezelést lehet megvalósítani és a felhasználó számára, amellyel a tervezés fázisában tudunk segédeszközt nyújtani.

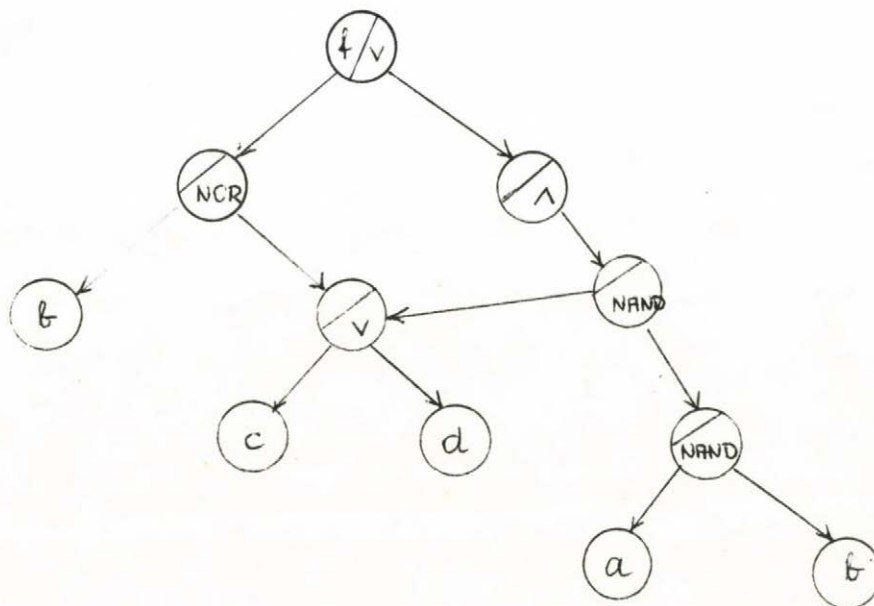
Ebben az esetben a Boole-függvények leírása lényegében a formulának megfeleltethető formulagráf, ami esetünkben egy olyan irányított gráffal való leírás, ahol a gráf csucsaihoz

rendelünk egy nevet és egy műveleti jelet /a név esetleg el is maradhat/. A csucsból kivezető élek a műveleti jelhez tartozó argumentumoknak megfelelő csucsokba mutatnak, amelyek vagy újabb műveleti jelek, vagy változók. Ezzel a leírással tetszőleges kombinációs hálózat megadható /lásd 2. ábra/.



$$f = \text{NOR}/b, c \vee d / \vee a \wedge \text{NAND}/c \vee d, \text{NAND}/a, b //$$

formulagráf:



2. ábra

A függvény kezelő rendszer inputja a függvény valamely formulája vagy a függvényt leíró implicit formularendszer. A műveletvégzéshez viszont legalább a függvény formulagráfjának ismeretét fel kell tételezni. Azt az előkészítő feldolgozó munkát, amelynek során a formulagráf előáll a programrendszer ismertetésekor részletezzük.

Ezek után áttérhetünk a Boole-függvény kezelő rendszerben megvalósított műveletek áttekintésére.

1., Azonos átalakítások

Ehhez a műveletcsoporthoz három minőségileg más művelet tartozik.

- a. A konverzió jellegű átalakítások. Ezek diszjunktív vagy konjunktív normálformában lévő függvényekre értelmezett műveletek. A függvény lehet formula alakban, helyértékes belső reprezentációban, formula gráf formájában vagy ki-tüntetett diszjunktív normálformában /igazságtábla/. A rendszer a különböző leírások közötti konverziót elvégzi.
- b. Egy függvény tetszőleges műveleteket tartalmazó formulájából a függvény egy diszjunktív normálformájának előállítás.

A függvény lehet formula alakban vagy formulagráf formájában. Az átalakítást a formulagráf alapján végzi a rendszer.

- c. A szintézis feladat néhány esete. Egy függvényt vagy egy függvényrendszert leíró adott követelményeknek eleget tevő nemredundáns vagy minimális formula illetve formularendszer előállítása. A függvény/ek/ megadhatók igazságtáblával vagy diszjunktív normálformájukkal. A szintézis elvégezhető $\{\wedge, \vee, \neg\}$ vagy $\{\text{NAND}\}$ vagy $\{\text{NOR}\}$ bázison. A szintézis eredménye lehet normálforma vagy zárójeles formula. A szintézis eljárások teljesen vagy parciálisan meghatározott függvényre vagy függvényrendszerre alkalmazhatók.

2., Függvények közötti műveletek

E műveletcsoport keretében az alábbi műveletek elvégzésére képes a rendszer:

- egy teljesen meghatározott függvény adott változója szerinti Boole derivált képzése
- szuperpozíció elvégzése
 - a függvény egy változója helyébe egy függvény behelyettesítése
 - a függvény egy /vagy több/ változója helyébe konstans(ok) behelyettesítése
 - a függvény értékének a változók adott érték kombinációja melletti kiszámítása

- függvények feletti Boole műveletek elvégzése
 - . függvény negálása
 - . függvény duálisának képzése
 - . függvények közötti többargumentumos Boole műveletek elvégzése /konjunkció, diszjunkció, NAND, NOR, antivalencia, stb./

A függvénykezelő rendszer realizálásánál az adatok számára egy kapcsolt listastruktúra látszott a legmegfelelőbbnek. Ezért választottuk a Digitális Osztályon kidolgozott LIDI 72 listastruktúrát az adatok tárolására. Ugyanakkor azon elképzelésünk megvalósítására, hogy jól megválasztott adatstruktúra esetén a feladatokat vagy azok nagy részét az adatstruktúrán értelmezett műveletekkel oldjuk meg.

A függvény kezelő rendszer LIDI 72-ben realizált adatstruktúráját és a függvénykezelő programrendszert a következőkben ismertetjük.

2. A programrendszer általános leírása

A rendszer adatait egy listastruktúrában a LIDI 72-ben ábrázoljuk. A feladatokat a listakezelés eszközeivel oldjuk meg. A LIDI 72-ben a listakezelés megszervezésére szubrutinok állnak rendelkezésre. A függvénykezelő rendszer e szubrutinok felhasználásával látja el feladatát.

A rendszer számítógépes realizációja az MTA SzTAKI CDC 3300-as számítógépére FORTRAN nyelven készült. A program kihasználja a CDC 3300 operációs rendszerének OVERLAY lehetőségeit.

A programrendszer feladatait két osztályba soroljuk

- adatstruktúra kialakító műveletek
- függvények feletti műveletek /függvénykezelés/

a. Adatstruktúra kialakító műveletek

Az input adatok alapján a listastruktúra előállítása vagy az input adatok elhelyezése egy már meglévő listastruktúrába.

A feladat megoldása során először a LIDI 72 bemenőnyelvén megadott információt a LIDI 72 INPT alprogramja elhelyezi a listastruktúrába. E feladat érdemi része a listába ily módon elhelyezett információ feldolgozása.

A feldolgozás jelenti a LIDI 72 adatterületén egy olyan struktúra létrehozását, amely biztosítja a 2-es osztályba tartozó műveletek végrehajthatóságát. A rendszernek ezt a részét teljesnek tekintjük.

b. A LIDI 72 adatterületen elhelyezett Boole-függvények leírását megadó strukturán végzett műveletek

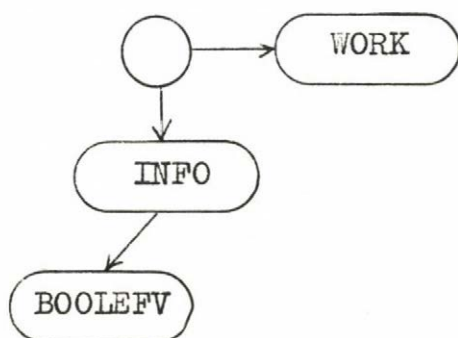
E műveletek a függvénykezelő rendszer függvények feletti műveletei.

A rendszernek ez a része bővithető a felhasználói igénynek megfelelően. E műveletek nem függetlenek az 1-es osztályba tartozóktól. Tehát csak akkor végezhetők el, ha az 1-es osztályba tartozó műveletekkel létrehozható teljes struktúra vagy e struktúra megfelelő kiépítettségben rendelkezésre áll.

2.1. A függvénykezelő rendszer adatstruktúrája

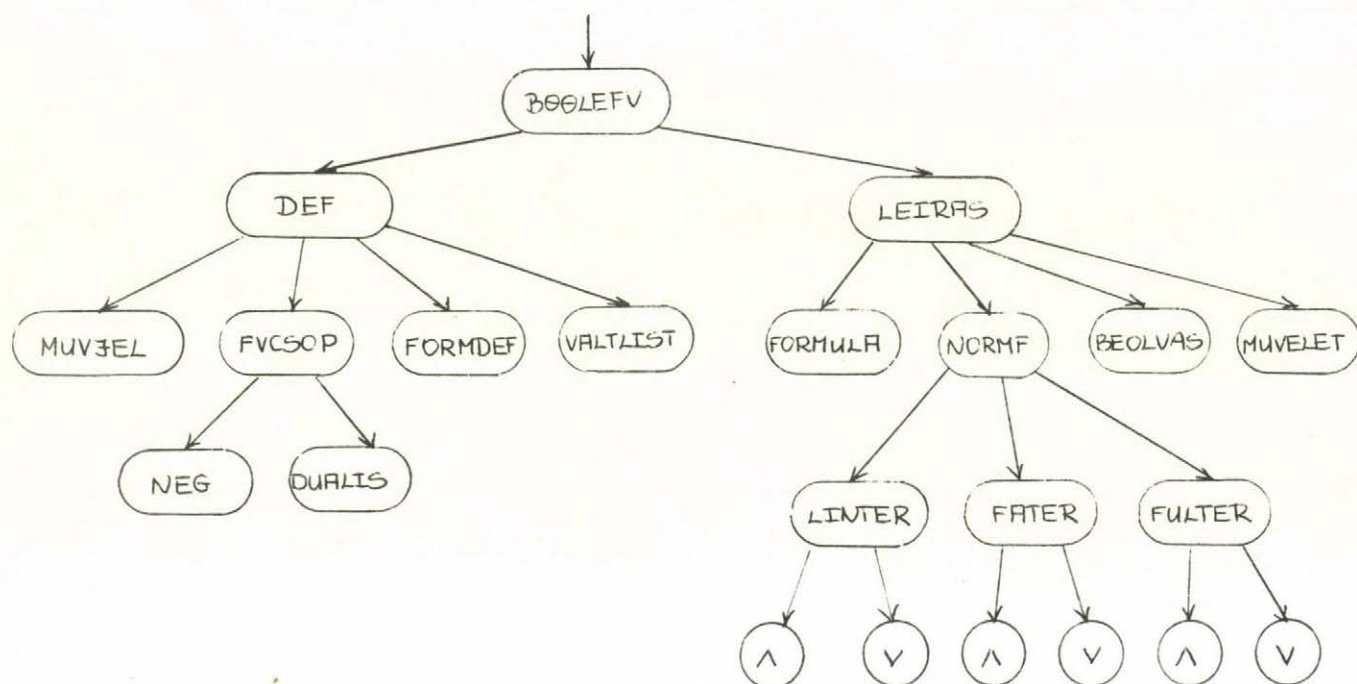
A következőkben a különböző formában megadott Boole-függvényeknek és a rájuk vonatkozó egyéb információinak a LIDI 72 listastruktúrába való beillesztésével foglalkozunk.

A listastruktúra "INFO"eleme /csucsa/ alá kapcsoljuk a "BOOLEFV" listaelemet /csucsot/ és e csucs alá kerül minden, a Boole-függvényekre vonatkozó információ. /3. ábra/



3. ábra

Az információt leírás és definíció jellegű részre osztjuk. Ennek megfelelően a struktúra inicializálása egy "gyökér" gráf kialakítását jelenti /4. ábra/, amely az összes információtypus elhelyezését biztosítja.



4. ábra

A "gyökér" gráf

Leírás jellegűek

- az egyes függvények normálformáinak helyértékes ábrázolásai. Ezek a NORMF szögpontra kerülnek.
- a függvények formulagráfjai, amelyek a FORMULA szögpontra helyezkednek el.
- a beolvasott függvény formulája, amelyet a BEOLVAS szögpontra helyez az INPT rutin.

- a Boole-függvények feletti műveleteket előíró információ vagy formula vagy fa gráf alakban, ami a MUVELET szögpontra kerül.

Definíció jellegűek

- a függvény változóinak listája a VALTLIST szögpontra alatt.
- a függvény minőségére / Φ -Boole-függvény esetén, a komponens függvények megadása/ vonatkozó információ és a negált, valamint a duális függvénypárok felsorolása a FVCSOP szögpontra alatt található.
- a műveleti jelek tartalmát kifejező információ a MUVJEL szögpontra alatt van.
- a formulagráfok kezelésére vonatkozó információ és az egy függvényhez tartozó különböző formulák létrehozásához szükséges információ a FORMDEF szögpontra alatt helyezkedik el.

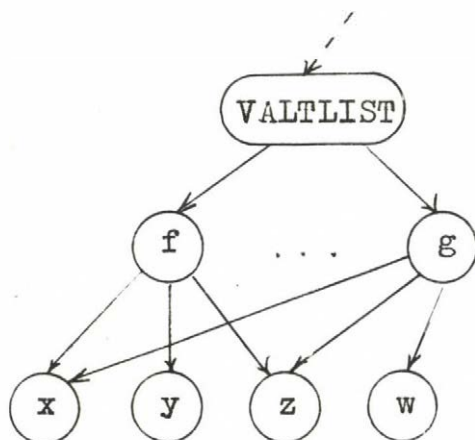
Az adatstruktúra állandó /gyökér/ része után ismertetjük a függvények különböző ábrázolásmódjainak realizálását és a köztük lévő kapcsolatokat megadó információ ábrázolását a struktúrában.

A továbbiakban a listastruktúra egy-egy részletét illusztráló gráfban egy szögpontra, amely mindig egy listaelem, sokszor a gépi reprezentációnak megfelelően részletezünk. Ekkor a rajzon szereplő téglalap egy 24 bites szót reprezen-

tál.

A formulák beolvasott alakja, mint karaktersorozat a BEOLVAS szögpontra alá kerül egy név nélküli listaelem attributumaként. A strukturakialakító program ezt az adatot használja fel, majd feldolgozás után törli a listából. A feldolgozás érdekessége, hogy veremtechnikával /a formula szintaktikázása után/ a formulából fordított lengyel /reverse polish/ formulát állít elő, amelynek alapján /mint az ismert [3]/ a formulagráf felépíthető.

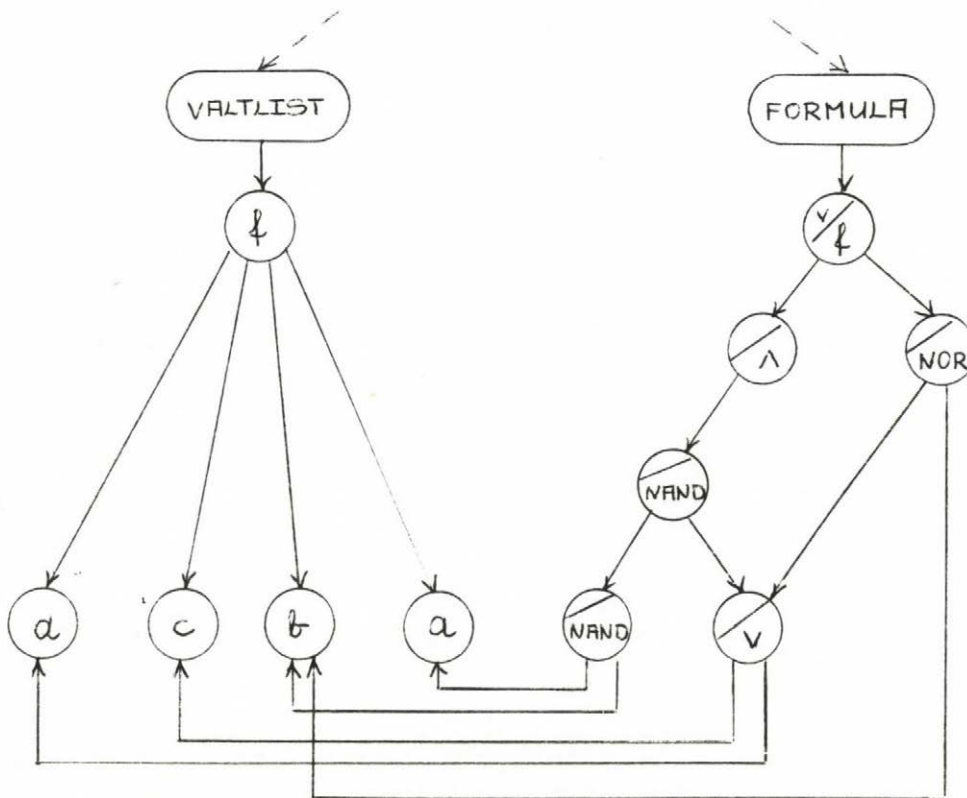
Minden beolvasott függvény nevével együtt a változók listája adott a VALTLIST szögpontra alatt. /5. ábra/



5. ábra

Az f/x,y,z/ és g/x,z,w/ változólistája

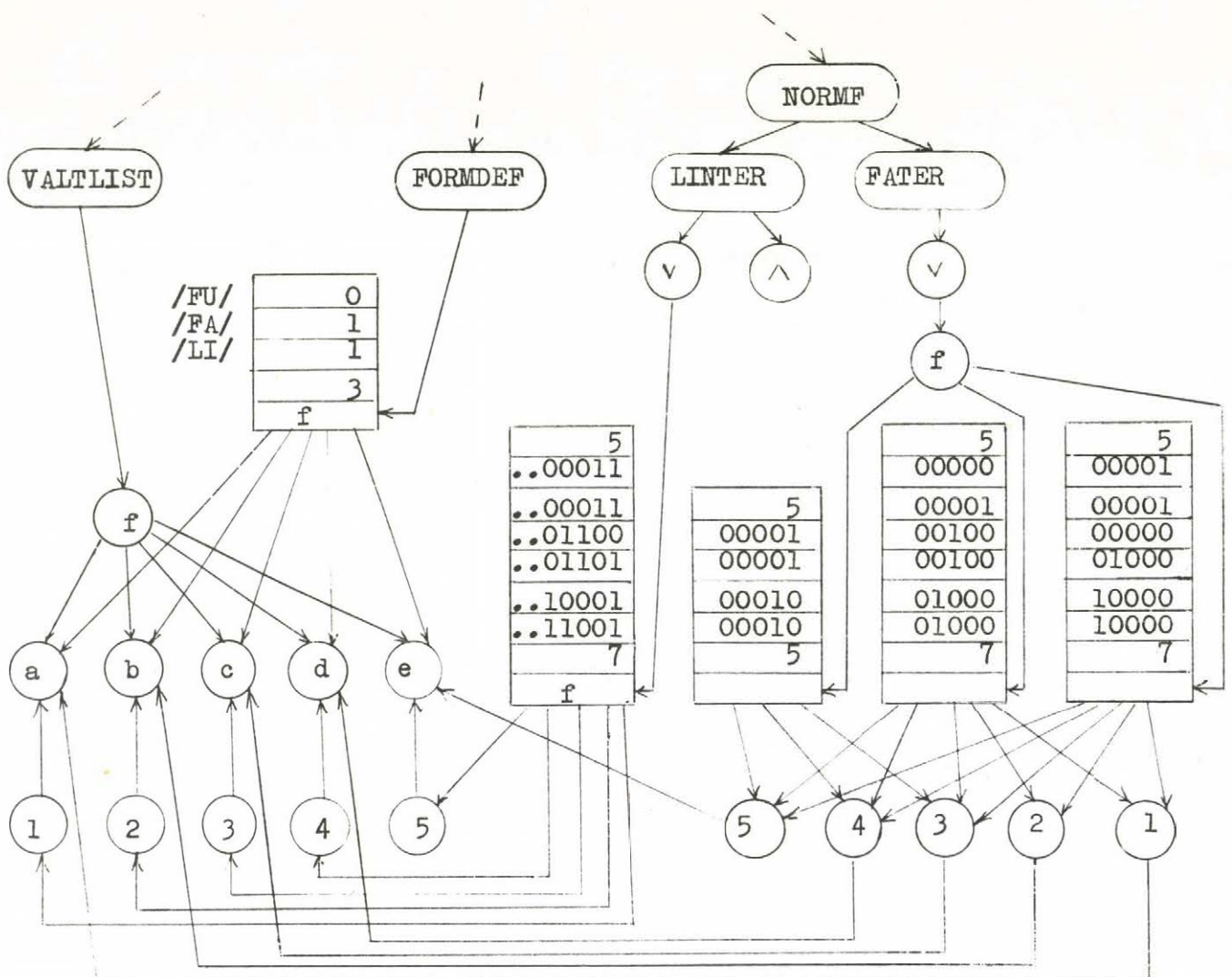
Az egyes formulák /formulagráf vagy normálforma alakúak/ szintén kapcsolódnak ehhez a változólistához. A 2. ábrán lévő formulagráf adatstrukturánkba való illesztésével illusztráljuk ezt a formulagráffal való ábrázolás esetében. /6. ábra/



6. ábra

A normálformában való ábrázolás az elemi termek helyértékes ábrázolását jelenti. A listastrukturában a termek listáját és a függvénynevet tartalmazó listaelemből egy i / $1 \leq i \leq n$, n a függvény változóiinak száma/ egész számot tartalmazó

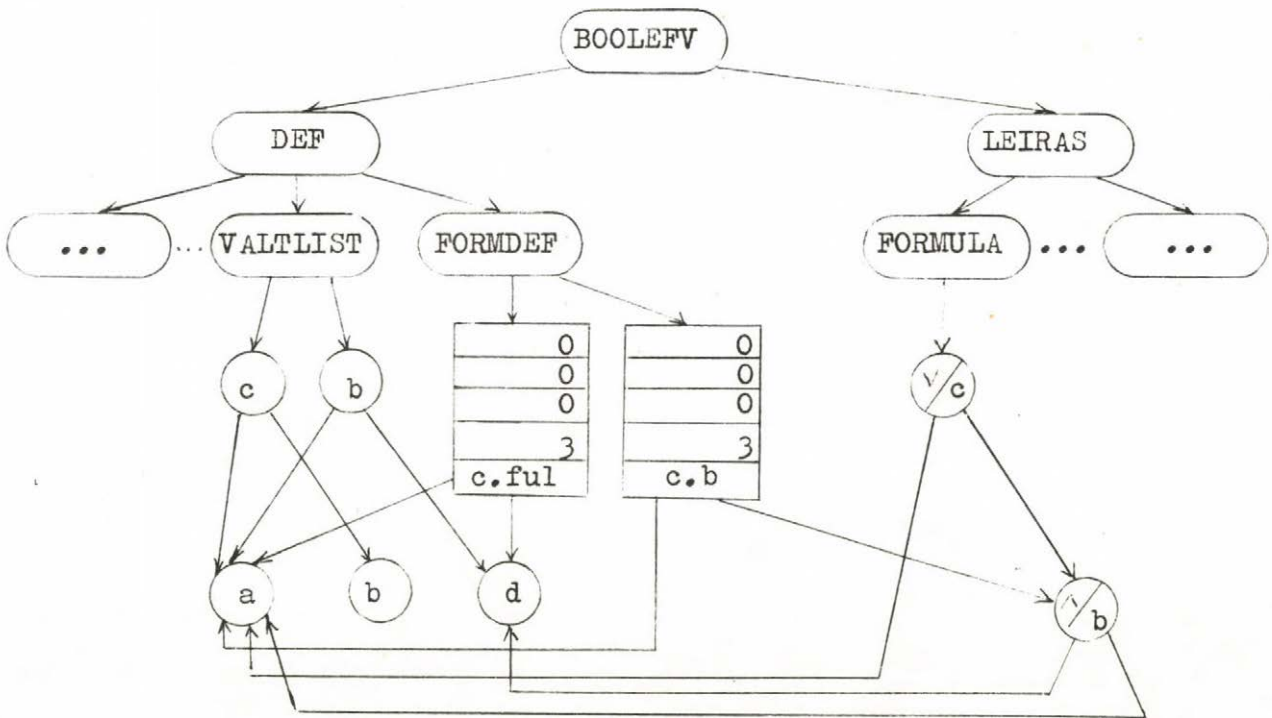
listaelemen keresztül jutunk a függvény i -edik változójához a VALTLIST alá /7. ábra/. A FORMDEF szögpont alatt az f szögpont első két argumentuma mutatja, hogy kétféle helyértékes ábrázolása van az f -nek.



7. ábra

Az $f=ab \bar{a}cd \bar{a}de$ helyértékes ábrázolásai

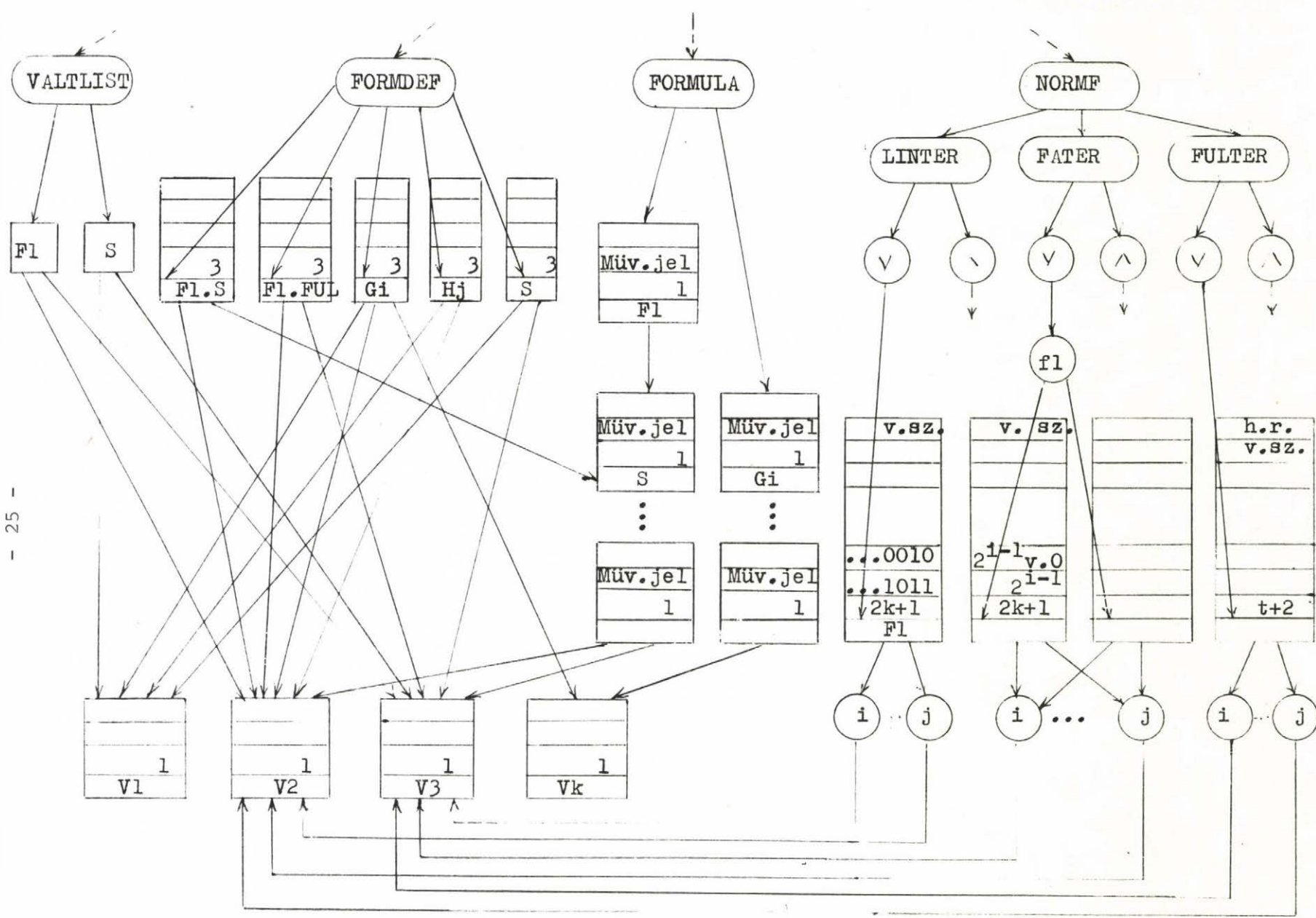
A FORMDEF alatt a formulagráfra vonatkozóan is van információ, ha a függvény formulagráfja szuperpozícióval állt elő. Így aztán a részfüggvények bármikor megtalálhatók. Legyenek az input függvények $c = a \vee b$, $b = a \wedge d$. Ekkor az adatstruktúra a 8. ábrán látható.



8. ábra

Az alábbiakban még részletezzük a formulamegadás, a változólista és a struktúra még nem tárgyalt részeinek szerkesztési szabályait.

A FORMULA kulcsszó alatt a függvény valamely "fordított lengyel formulájával" ekvivalens gráfot tárolunk. A gráfban lévő listaelemek szerkezete /lásd 9. ábra/ fej, két attributum, pointerek. Az első szinten lévő fejben a függvényazonosítót, az első attributumban a függvényformulában legmagasabb szinten lévő műveleti jelet tároljuk, a második attributum műveletvégzések részeredményeinek tárolására szolgálhat, a pointerek száma az attributumbeli operátor argumentumai számával egyezik meg. A pointerek az argumentumoknak megfelelő listaelemek fejére mutatnak. Ezen listaelemek bármelyike vagy az előzőekben részletezett listaelemmel azonos szerkezetű, azzal a változtatással, hogy a fejben lévő információ vagy üres, vagy egy azonosító. Az első esetben az első attributumban egy műveleti jel van, a második esetben az első attributumban akkor van műveleti jel, ha a listaelem a formulagráf belső eleme és az első attributumban nincs műveleti jel, esetleg üres, ha a listaelem a függvény valamely független változóját írja le.



9. ábra

A "VALTLIST" szögponthoz kapcsolódó szerkezete a következő /lásd 9. ábra/. A VALTLIST szögponthoz kapcsolódó szinten olyan listaelemek helyezkednek el, amelyek fejében az egyes függvények azonosítói találhatóak. Attribútumuk nincs. A pointerok a függvény változóira mutatnak. A függvény változóit reprezentáló egyébként terminális listaelemek szerkezete a következő. Két attribútumuk van, az első a helyértékes reprezentáció előállítását biztosítja, a második egyes műveleteknél munkarekesz, a fejben a változónevet tároljuk.

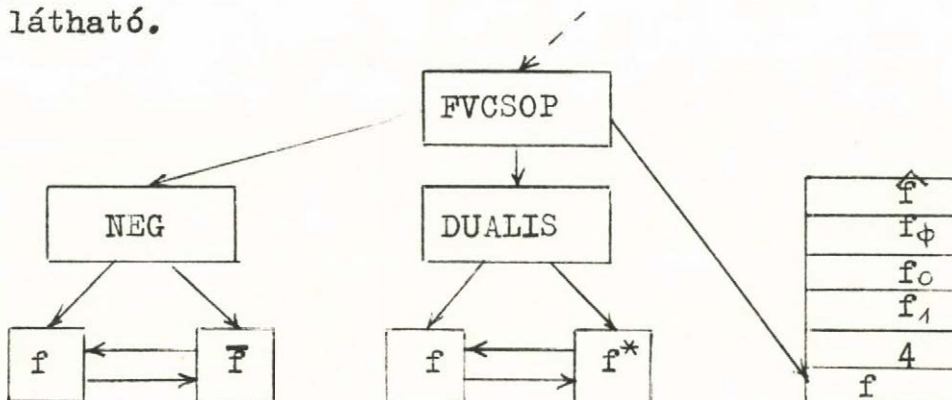
A "LINTER" szögponthoz kapcsolódó "AND" vagy "OR" szögponthoz kapcsolódó szinten lévő attribútumos nem terminális listaelem írja le a függvényt. A listaelem fejében a függvényazonosítót, az attribútumban a normálformában szereplő elemi kifejezések I, II komponenseit és a változók számát tároljuk. E fejhez tartozó pointerok száma a függvény változóinak számával egyezik meg. E pointerok egyetlen pointerrel rendelkező attribútum nélküli listaelemekre mutatnak.

Egy ilyen listaelem fejében az $i=1,2,\dots,22,23$ számok lehetnek, a pointer egy változót leíró listaelemre mutat. Az érintett függvény elemi kifejezései listájában az a változó, amelyre a pointer mutat a 2^{i-1} helyértéknek felel meg.

E leirási mód mellett egy másik helyértékes leírást is használunk a Boole-függvények közötti számítások megkönnyítése céljából. A "FATER" szögpontra kapcsolódó "AND" vagy "OR" szögpontok alatti listaszervezet a következő. Az első szinten lévő listaelem fejében a függvények nevét találjuk. Egy ilyen listaelem pointerai a normálforma elemi kifejezéseit a "LINTER" szögpont alatti módon leíró listafejekre mutatnak.

A függvények teljes /vagy kitüntetett/ normálformáinak helyértékes reprezentációját a "FULTER" szögpont alatti első szinten lévő "AND" vagy "OR" szögponthoz kapcsolódó nem terminális attributumos listaelem írja le. E listaelem szerkezete a "LINTER" alattiétól abban különbözik, hogy a II komponensek hiányoznak, és a változók számán kívül a függvény helyértékes reprezentációját is tartalmazza az attributum. Egyébként mindenre a "LINTER"-nél elmondottak érvényesek.

Az "FVCSOP" szögpont alatti listaszervezet a 10. ábrán látható.

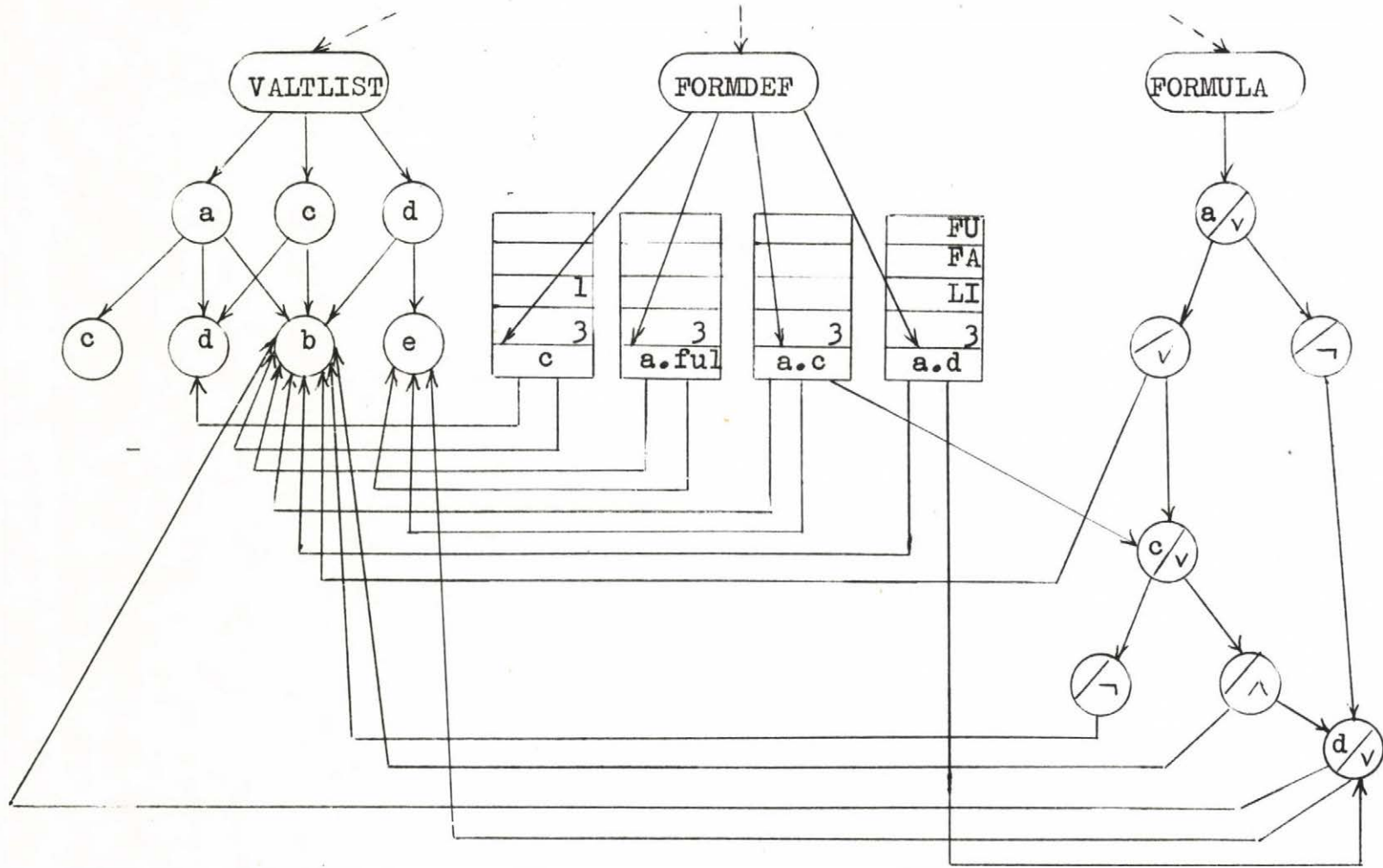


10. ábra

A "FORMDEF" szögponthoz olyan formulák változólistáit adjuk meg, amelyek formulagráfjából azonos átalakítással jöttek létre.

E formulák lehetnek:

- a. Egy szimultán egyenletrendszerrel megadott f függvény teljes behelyettesítés utáni formulája. Ebben az esetben a "FORMDEF" alatti, az f .ful nevet tartalmazó 3 attributumú szögponthoz található az f változó /amelyek egyébként a "VALTLIST" alatti terminális elemek/. /Ha behelyettesítés nincs, akkor a FORMDEF alatt csak abban az esetben szerepel ez a fej, ha a függvénynek legalább egy helyértékes formáját már előállítottuk a listában./
- b. A teljes behelyettesítéssel kapott formula valamely szükítése, amelyben bizonyos behelyettesítéseket nem hajtunk végre. Ebben az esetben a "FORMDEF" alatti 3 attributumú szögponthoz, az f függvény e formulájához az $f.g_1 \dots g_n$ nevet rendeljük, ahol g_1, g_2, \dots, g_n a nem behelyettesített függvények, amelyek e függvénynek változói, amelyek az f .ful formulába behelyettesített belső függvények nevet tartalmazó, a formulagráfban lévő listafejek. Ezenkívül a függvény változói között szerepelhetnek a "VALTLIST" alatti terminális elemként ábrázolt változók is.
/lásd 11. ábra/



$$\begin{aligned}
 a &= b \wedge c \vee \bar{d} \\
 c &= \bar{b} \vee d \wedge b \\
 d &= b \vee e
 \end{aligned}$$

$$\begin{aligned}
 a.c &= b \wedge c \vee \overline{b \vee e} \\
 a.d &= b \wedge \bar{b} \vee d \wedge b \vee \bar{d} \\
 a.ful &= b \wedge \bar{b} \vee \bar{b} \vee e \wedge b \vee \overline{b \vee e}
 \end{aligned}$$

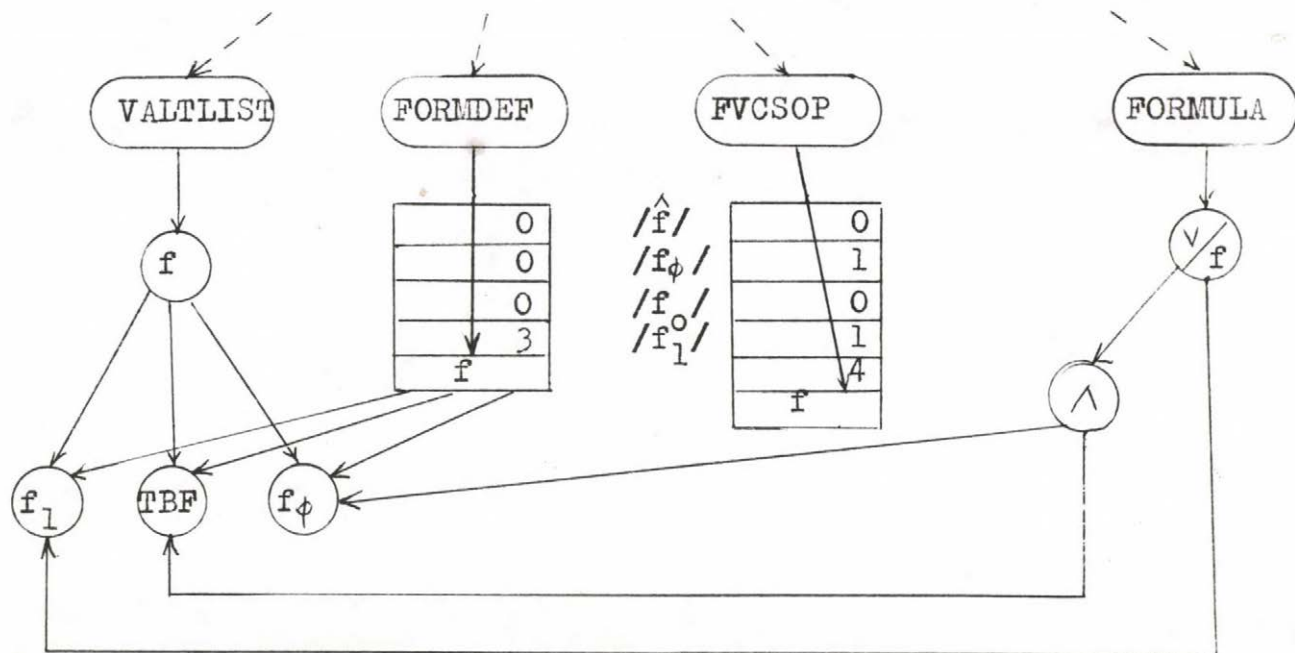
Belső függvények nevei vagy formuláinak nevei is szerepelhetnek a FORMDEF alatt a, vagy b, formában is, ha azok a azámitás során szerepeltek már önállóan.

A FORMDEF alatti első szinten lévő fejek három attributumában a formula helyértékes reprezentációinak létét jelezzük. Az első szó linter, a második szó fater, a harmadik szó a fulter jellegű reprezentációra vonatkozóan ad felvilágosítást. Ha valamelyik attributumbeli szó 0, negatív vagy műveleti jel /OR,AND/, akkor a megfelelő alakú reprezentáció nem létezik, létezik mindkét /OR,AND/ főműveletre, vagy létezik a szóban feltüntetett főműveletére.

A programrendszer a ϕ -Boole-függvény megadására az

$$f = f_1 \vee \gamma f_\phi \quad \text{vagy az}$$
$$f = f_1 \vee \gamma \bar{f}_0$$

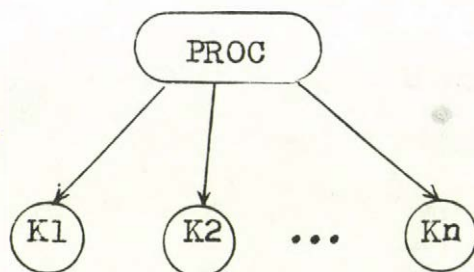
alaku leírást fogadja el, ahol a γ tetszőleges Boole-függvény jeleként a rendszerben kitüntetett, TBF azonosító kell hogy szerepeljen. A beolvasást követő strukturalakítás eredményeként a FVCSOP szögpont alá kerül az az információ, hogy az f függvényt milyen módon adtuk meg. Például, ha az adat $f=f_1 \vee TBF \wedge f_\phi$, akkor az adatstruktúrát a 12. ábra mutatja.



12. ábra

A függvények feletti művelet létezését és a strukturakialakító műveletek elvégzését a BOOLEFV szögpont alá bevitt "PROC" gráffal írjuk elő. /13. ábra/

A strukturakialakítás után a PROC gráfot töröljük a listából. A gráfban a K_1, K_2, \dots, K_n az elvégzendő műveleteket jelölik. E műveletekkel a program leírása során még foglalkozunk majd.



13. ábra
A PROC gráf

A Boole-függvényeken elvégzendő műveleteket a MUVELET szögpont alatt formulával vagy fa gráffal adhatjuk meg.

A műveletek elvégzésére azaz a műveletvégző OVERLAY szegmen- sek aktivizálására három lehetőség van. A leggyakoribb műveletek elérését az OPN/20/ tömbbe elhelyezett műveleti jelek e tömbbeli indexe biztosítja.

Az OPN tömb tartalma:

index	1	2	3	4	5	6	7	8	9	10	11
OPN	NEG	AND	OR	NAND	NOR	XOR	IMPL	EKV	MIN	EGYSZ	RED

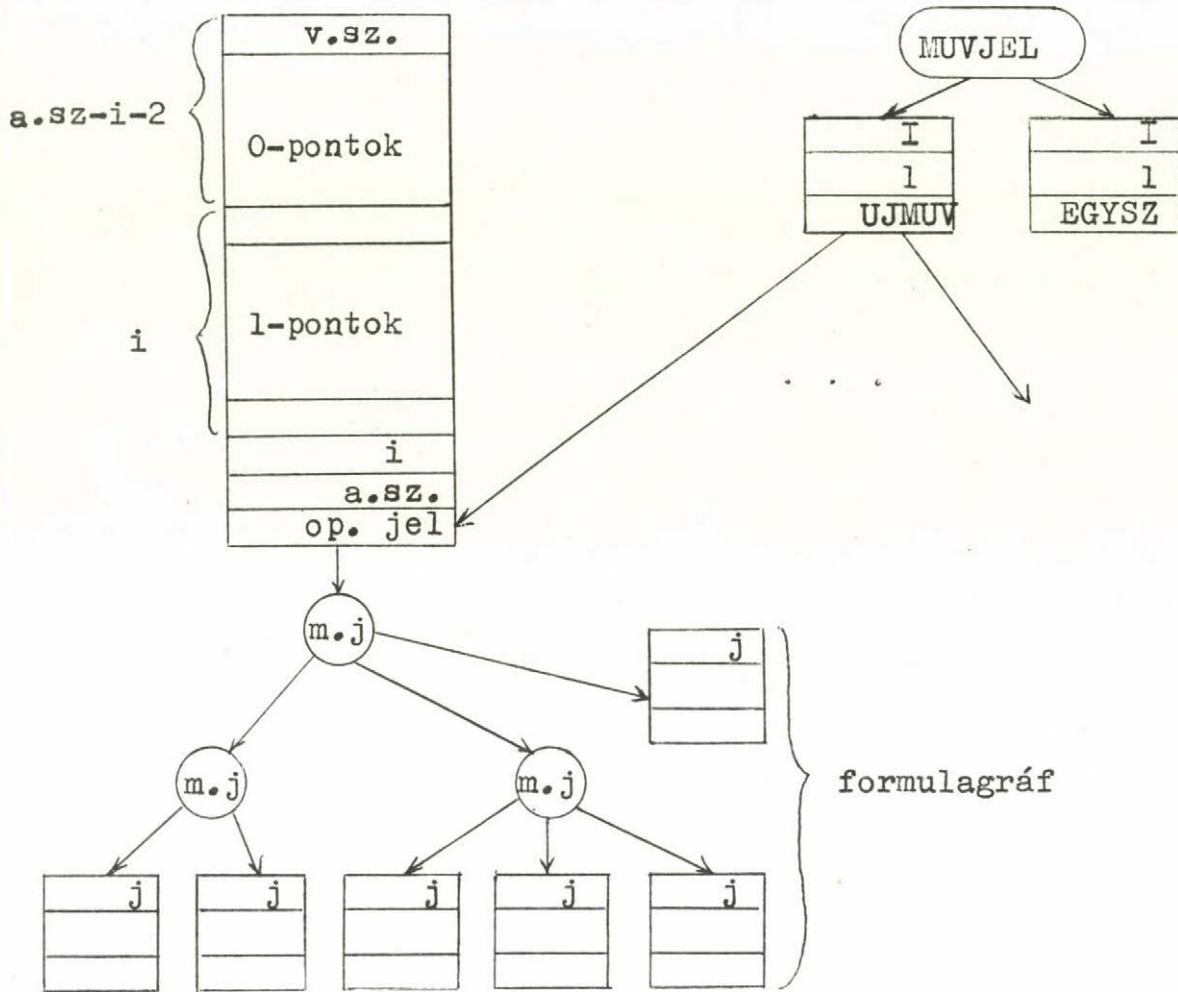
index	12	13	14	15	16	17	18	19	20
OPN	BDIFF	NANDS	NORS	ORANDZ	HERT	CHELY	SZINT	EGYMO	DUAL

A kevésbé gyakori műveletek elvégzése a MUVJEL szögpon-
t alatti egyattributumos, terminális, a fejben a művelet
azonosítóját tartalmazó szögpont attribútuma alapján szer-
veződik. /14. ábra/

Az a törekvésünk, hogy a Boole-függvények feletti operá-
ciók rendszere új műveletekkel bármikor bővithető legyen.
Ennek érdekében minden új műveletet, amelyet lehet, egysé-
ges elvek szerint definiálunk klasszikus műveleteket tar-
talmazó formájukkal és /vagy/ meghatározott szerkezetű
igazságtáblájukkal.

Igy ezek az új operációk definíciójuk alapján egyféle al-
goritmussal aktivizálhatók.

A definícióhoz a MUVJEL szögpon-
t alá bevezetjük az UJMU-
V egyattributumos szögpon-
tot, ahol az attribútum adja az
információt a program futtatásához. E szögpon-
t alatt talál-
hatók a definíciók. Az UJMU-
V szögpon-
t alatti szinten attri-
butumos terminális vagy nem terminális fejek vannak. A fej-
ben találjuk a művelet nevét, az attribútumban az igazság-
táblát, a szögpon-
t alatt a formulagráfot. /14. ábra/



14. ábra

A formulagráfban a változók név nélkül szerepelnek és attribútumukban 0 áll, ha a változók sorrendje mellékes és $i=1,2,\dots$, ha az illető változó az asszimmetrikus függvény i -edik változója.

Az igazságtábla szerkezete. Az első attribútumban az $i \geq 0$ szám áll. Ha nem teljesen meghatározott a függvény, akkor

i az l -pontok száma. Ha $i=0$, akkor teljesen meghatározott függvényről van szó, és az igazságtábla csak az l -pontokat tartalmazza. A pontok helyértékes reprezentációiban szerepelnek, a helyérték kiosztása tetszőleges, szimmetrikus esetben $/j=0/$, aszimmetrikus esetben $/j\neq 0/$ pedig a 2^{j-1} megfeleltetés áll fenn.

2.2. Adatstruktúra kialakító műveletek

- MUVELET

A művelet végrehajthatóságának feltétele a gyökér MUVELET nevű szögpontja alatt elhelyezett műveletleírás. A művelet hatására, ha a műveletleírás formula alakban adott, akkor a program végrehajtja annak szintaktikai vizsgálatát, majd az első osztályba tartozó műveletek elvégzése után aktivizálja a Boole művelet szervező BOOLOP szubrutint, amely a 2-es osztályba tartozó műveletek elvégzését szervezi a MUVELET szögpont alatti információ alapján.

- FORMULA

A művelet végrehajthatóságának feltétele a BEOLVAS nevű szögpont alatt kiegészítő formátumban [2] megadott információ jelenléte.

A FORMULA művelet végrehajtása a következőt jelenti.

Elvégzi a kiegészítő formátumban megadott információk, Boole-függvény rendszereket leíró formulák szintaktikai vizsgálatát.

A szintaktikusan helyes információ alapján elkészíti a Boole-függvények formulagráfjait fa alakban és a FORMULA szögpontra alá kapcsolja azokat.

- VALTLIST

A művelet végrehajthatóságának feltétele vagy ugyanaz mint a FORMULA műveleté, vagy a FORMULA művelet eredménye.

A VALTLIST művelet végrehajtása a következőt jelenti.

Ha a FORMULA műveletet még nem hajtottuk végre, akkor előbb azt aktivizáljuk. Ezután a formulagráfok alapján elkészülnek a Boole-függvények változólistái úgy, hogy a formulagráfok és a változólisták azonos nevű szögponthoz egyetlen szögpontra ábrázolja.

- FORMDEF

A művelet végrehajthatóságának feltétele, hogy a VALTLIST művelet végrehajtása megtörténjen azokra a függvényekre, amelyekre a műveletet alkalmazni kívánjuk.

A végrehajtás eredménye a FORMULA szögpontra alatt leírt Boole-függvény rendszer szimultán helyettesítése. A szimultán helyettesítés a formulagráfok következő transzformációját jelenti.

Ha valamely Boole-függvény neve változóként is előfordul, akkor a változó szögpontra mutató él helyére a függvényt leíró formula gyökérszögpontjára mutató él kerül. A gyökérszögpontot a FORMULA szögponttal összekötő él törlődik. Ezután létrehozuk a FORMDEF szögpont alatt a behelyettesítéssel nyert új függvények változólistáit és generáljuk az új függvények neveit.

Ha az új függvényt az összes lehetséges helyettesítés elvégzésével kaptuk F-ből, akkor a generált név F.FUL lesz. Ha az F-be behelyettesíthető függvények közül a G_1, G_2, \dots, G_N -et nem helyettesítjük be, akkor a generált név $F.G_1.G_2.\dots.G_N$ lesz.

- FVCSOP

A művelet végrehajthatóságának feltétele a VALTLIST művelet végrehajtása.

A FVCSOP művelet végrehajtása jelenti a Φ -Boole-függvényekre vonatkozó információk elhelyezését a FVCSOP szögpont alá.

2.3. Függvények feletti műveletek

Azonos átalakítások

A konverzió jellegű átalakítások és a tetszőleges formulából diszjunktív vagy konjunktív normálforma előállítására a HERT művelet megfelelő paraméterekkel való aktivizálásával történik.

- HERT/ a_1, a_2, \dots, a_n /

Az a_1, a_2, \dots, a_{n-1} paraméterek függvénynevek, amelyek egyszerű vagy összetett azonosítók. Az a_n az AND vagy az OR szimbólum. A művelet végrehajthatóságának feltétele, hogy egyszerű függvénynevek a VALTLIST szögpontra alatt az összetett függvénynevek a FORMDEF szögpontra alatt szerepeljenek a listában.

A művelet az adott Boole-függvényrendszer a_1, a_2, \dots, a_{n-1} elemeinek belső reprezentációit készíti el a gyökér LINTER nevű szögpontra alatt elhelyezkedő OR szögpontra alá, ha $a_n = \text{OR}$ és az AND szögpontra alá, ha $a_n = \text{AND}$. Az OR szögpontra alá a formulával ekvivalens diszjunktív normál forma, az AND alá a konjunktív normál forma kerül.

Szintézis jellegű átalakítások

- MIN/ a_1, a_2, \dots, a_n /
- EGYSZ/ a_1, a_2, \dots, a_n /

Az a_1, a_2, \dots, a_n paraméterek egyszerű vagy összetett függvénynevek. A végrehajtás feltétele ugyanaz mint a HERT műveleté. A MIN és az EGYSZ művelet a függvény $/1 \leq n \leq 23/$ szimultán minimalizálását vagy egyszerűsítését végzi el diszjunktív normálformában.

- ORANDZ/ $f_1, \hat{f}^{\#}, M/1/, M/2/, MZ/$
- NANDS/ $f_1, \hat{f}^{\#}, M, MZ/$
- NORS/ $f_1, \hat{f}^{\#}, M, MZ/$

Egy f nem teljesen meghatározott Boole-függvény közel optimális zárójeles szintézisét végzi el $\{NOR\}, \{NAND\}$ vagy $\{\wedge, \vee, \neg\}$ bázison a műveletek változói száma és a zárójelmélység korlátozása mellett. Az eljárások közös elve a [4]-ben található. A szintézis eredménye sornyomtatón jelenik meg. A paraméterek: f_1 a függvény alsó határa, $\hat{f}^{\#}$ a függvény felső határának duálisa, M a NOR ill. NAND, $M/1/$ az OR, $M/2/$ az AND maximális bemenetszáma, MZ a maximális zárójelmélység. A program a két $/f_1, \hat{f}^{\#}/$ függvény DNF-jának helyértékes reprezentációjával dolgozik.

Függvények közötti műveletek

- /BDIFF/f/ $fx_{i_1}, x_{i_1}, \dots, fx_{i_k}, x_{i_k}$ ///

A végrehajtás feltétele, hogy az f függvény formulagráfjával és változólistájával adott legyen. Az utasítás hatására az f függvény

$$\frac{df}{dx_{i_1}}, \frac{df}{dx_{i_2}}, \dots, \frac{df}{dx_{i_k}}$$

deriváltjait DNF-ban helyértékes ábrázolásmódban előállítja és $fx_{i_1}, fx_{i_2}, \dots, fx_{i_k}$ néven a LINTER, OR alá elhelyezi ezeket.

- FORMDEF

A művelet a VALTLIST strukturaépítő művelet után a FORMULA szögpont alatt lévő függvényekre elvégzi az összes lehetséges szuperpozíciót. Lásd FORMDEF mint strukturaépítő művelet.

Függvények feletti Boole műveletek

- CHELY/ $a_1, a_2, b_1, c_1, b_2, c_2, \dots, b_n, c_n$ /

Az a_1, a_2 paraméter függvénynév, a b_1, b_2, \dots, b_n paraméterek az a_1 függvény változóinak nevei, a c_1, c_2, \dots, c_n Boole konstansok $/1 \leq n \leq k$, k az a_1 függvény változóinak száma/.

Az igaz értéket az EGY, a hamis értéket a NULL szimbólum jelöli. A művelet végrehajthatóságának feltétele, hogy az a_1 függvény formulagráfja és változólistája szerepeljen a listában.

A művelet eredménye, ha $a_1 \neq a_2$, az a_2 nevű függvény formulagráfja és változólistája. Ezt a függvényt az a_1 nevű függvényből úgy nyerjük, hogy a paraméterlistában szereplő b_i változók helyébe rendre a c_i konstansokat helyettesítjük. Ha $a_1 = a_2$, akkor a függvény minden változójának szerepelnie kell a paraméterlistában. Ekkor a művelet kiszámítja az a_1 függvény értékét, amelyet az a_2 változó vesz fel.

- DUAL/ da_1, a_1 /

Az a_1 teljesen meghatározott függvény da_1 duálisát helyértékes alakban állítja elő.

Az alábbi műveletek hatására a paraméterlistában szereplő a_1, a_2 függvények formulagráfjából elkészíti egy új formulagráfbeli szögpont bevezetésével az f eredmény formulagráfját.

- NEG/ f, a_1 /

- OR/ f, a_1, a_2 /, AND/ f, a_1, a_2 /, NOR/ f, a_1, a_2 /, NAND/ f, a_1, a_2 /,
XOR/ f, a_1, a_2 /

3. A realizált programrendszer

3.1. A műveletszervezést és a strukturakialakítást biztosító programok

A rendszert szervező program a 0-ás sorszámú OVERLAY lapon helyezkedik el. E program vezérlő része az OVERLAY/0/ főprogramja /"PROGRAM SZERVO"/, amely az IDIL27/IROOT/ szubrutin CALL IDIL27/I/ alaku hívásával biztosítja a műveletvégrehajtó rész aktivizálását. Az I=0,1,2 lehet. /16. ábra/

- a. Ha I=0, akkor a programrendszerhez hozzá kell rendelni azt a file-t, ahol a "gyökér"-rel inicializált lista található, /file név: TASKNAME, dsi=1111/ 4. ábra. Ez az inicializált lista ezen kívül tartalmazza a Boole-függvényekkel végezhető legfontosabb műveletek külső neveinek megfelelő belső neveket is /1. táblázat/.
- b. Ha I=1, akkor a teljes lista-inicializálást el kell végezni. Ebben az esetben a LIDI 72 bemenő nyelvén a 15.a, 15.b ábrán lévő adatmegadás szükséges.
- c. Ha I=2, akkor egy tetszőleges file-on lévő esetleg már korábbi munkák eredményeit is tartalmazó listát /dsi=8000/ használ a program. Ilymódon lehetőség van egy megkezdett munka folytatására.

A 8-as sorszámú lapon lévő program a feladat végrehajtása után a LIDI 72 LWRITW nevű rutinjának aktivizálásával a dsi=8000 azonosítóju file-ra kiírja a teljes listát és az eredmény listát a sornyomtatón is megjelenti.

SUBROUTINE IDIL27/IROOT/

A Boole-függvény kezelő rendszer /továbbiakban BFKR/ működését szervezi.

1. Bemenő paraméterének értéke alapján /mint az előzőkben láttuk az I paraméter értéke 0,1,2 lehet/ file-ról behozza a gyökér listát vagy az operatív tárban elkészíti a gyökér listát /60-as overlay lap/ vagy file-ról beolvas egy korábban létrehozott listát.
2. Beolvassa a LIDI 72 bemenő nyelvén megadott adatokat.
/11. szegmens/
3. Megvizsgálja, hogy az adatok biztosítják-e a BFKR helyes működését. /66-os overlay lap/
4. Elvégzi a listán a PROC gráffal definiált adatstruktúra kialakító műveleteket. /61.,62.,63. overlay lap/
5. Informálja a felhasználót a függvények feletti műveletek végrehajtása előtt a lista állapotáról. /67. overlay lap/

6. Kigyűjti a közös munkaterület meghatározott helyére a listában szereplő összetett neveket. /68. overlay lap/
7. Kikeresi a BFKR gyökérszögpontjainak, a beolvasás után megváltozott címeit. /CIMKER rutin/
8. Végrehajtja a listában definiált függvény-műveleteket. /BOOLOP szubrutin/

Az IDIL27/IROOT/ szubrutin által aktivizált szegmensek

E szegmenseket és kapcsolódásukat a 16. ábra mutatja.

- a. OVERLAY/60/. A lap vezérlőprogramja aktivizálja a GYOKEL szubrutint, amely a következő feladatokat hajtja végre.
 - A 15.a ábrán lévő adatot elhelyezi az üres listába az INFO szögpont alá. Az elhelyezéskor a külső nevekhez belső nevek rendelődnek. Mivel a belső név függ a beolvasás sorrendjétől, így biztosítjuk, hogy ezekhez a külső nevekhez a lista bármely aktivizálásakor ugyanazon belső nevek tartozzanak.
 - Törli az INFO alá bevitt fát.
 - A 15.b ábrán leírt fát elhelyezi az INFO szögpont alá.
 - Kiviszi az inicializált listát a dsi=l111 file-ra.

b. SEGMENT/11/

Aktivizálja a LIDI 72 INPT rutinját a BFKR adatainak beolvasására.

c. OVERLAY/66/

A lap vezérlőprogramja a PROGRAM INDIR. Végrehajtásának eredménye a gyökérhez tartozó szögpontok címeinek és belső neveinek, továbbá a műveletek belső neveinek közös munkaterületre való elhelyezése /INDIT rutin, 2.,3.,4. táblázat/ és a struktura műveletek végrehajtását vezérlő munkamező kitöltése.

d. OVERLAY/61/

A lap vezérlőprogramja a PROGRAM LIBULI.

- Kikeresi és a közös munkaterületre elhelyezi a gyökérszögpontok címeit. /CIMKER szubrutin/
- Végrehajtja a BEOLVAS ill. MUVELET szögpont alatt a LIDI 72 bemenő nyelven kiegészítő formátumban megadott információ /Boole-függvények halmazának leírása ill. Boole művelet generálás/ szintaktikai vizsgálatát. /VLISTI szubrutin/
- Ha a kiegészítő formátumban megadott információ szintaktikailag helyes, akkor a BEOLVAS szögpont alatt le-

irt Boole-függvényeket leíró formulagráfokat és a változólistákat elhelyezi a FORMULA ill. VALTLIST szögpontra alá /VLISTI szubrutin/, egyébként leállítja a program működését.

e. OVERLAY/62/

A lap vezérlőprogramja a PROGRAM BEBULI.

- A közös munkaterületre beírja a gyökérszögpontra cimeit. /CIMKER szubrutin/
- Elvégzi a Boole-függvénykezelő rendszer elemeinek összes lehetséges helyettesítését, és a helyettesítéssel nyert új függvények változólistáit elkészíti a FORMDEF szögpontra alatt. /SIEQUS szubrutin/ Ha a feladat nem végezhető el, hibajelzést ad és visszatér az IDIL27 rutinba.

f. OVERLAY/63/

A lap vezérlőprogramja a PROGRAM FIBERA.

- A közös munkaterületre beírja a gyökérszögpontra cimeit. /CIMKER szubrutin/
- Kikeresi a formulagráfok közül a Φ -Boole-függvényeket és a függvény leírását elhelyezi a FVCSOP szögpontra alá. /FIBULI szubrutin/

g. OVERLAY/67/

A lap vezérlőprogramja PROGRAM INFORM.

Megvizsgálja a BFKR adatstrukturáját és arról üzeneteket küld.

Az üzenetek a következők:

1. Kiírja a VALTLIST szögpontra alatt található változólisták száma n_{v1} és a FORMULA szögpontra alatt található formulagráfok száma n_f alapján a csak formulagráffal rendelkező formulák számát $k_1 = n_f - n_{v1}$ -et és azon formulagráfok számát, amelyeknek már van változólistája, $k_2 = n_{v1}$ -et a következő üzenettel:

*****A LISTAABAN k_1 FA EES k_2 HAALO VAN.*****

2. Ha a VALTLIST szögpontra terminális, kiírja a

*****NINCS VAALTOZOLISTA.*****

és ha a szögpontra nem terminális, a

*****VAN VAALTOZOLISTA.*****

üzenetet.

3. Ha a FVCSOP szögpont alatt nem talál Φ -Boole-függvényt, vagy a FORMDEF alatt behelyettesítéssel nyert függvényt leíró szögpontot, akkor a következő szöveget írja ki:

*****AZ FDEF SZOEGPONT ALATT NINCS VAALTOZOLISTA
VAGY NEM TOERTENT FIBOOLE FUEGGVEENY ADMINISZTRAACIO.*****

h. OVERLAY/68/

A lap vezérlőprogramja aktivizálja az összetett neveket közös munkaterületre kiíró eljárást.

i. SUBROUTINE CIMKER

Elvégzi a /CIMES/ közös adatmező /2. táblázat/ feltöltését a BFKR gyökeréhez tartozó szögpontok fejcímeivel.
/15b.ábra/

j. SUBROUTINE BOOLOP

A szubrutin a MUVELET szögpont alatt elhelyezett formulával vagy fával leírt műveletek elvégzését biztosítja.

- előállítja azt a formájában egységes információt, amely szükséges a műveletvégző rutinok működéséhez.
Ez az információ a következő:

- . a művelet argumentumainak száma. Ezt az IARG változó értékeként adja a BOOLOP a COMMON területén.

. az argumentumok azonosításához szükséges információ.
Ezt a COMMON területen a RES tömbbe dolgozza ki a
rutin a RES/IRES/-RES/IRES+IARG-1/ helyekre.

Az információ értelmezése a következő:

ha $RES/I/ > 0$, akkor ott az argumentum neve,

ha $RES/I/ < 0$, akkor a listabeli címének -1-szerese áll.

- aktivizálja a megfelelő műveletvégző rutint

Ez az aktivizálás a művelet jeléhez hozzárendeli azt a
számot, amely a műveletvégző programot tartalmazó
OVERLAY lap sorszám. A 2.1. pontban elmondottakkal
összhangban, ha OPN/I/ tartalmazza a műveleti jelet,
akkor ez a szám az I index, ha a műveleti jel a MUVJEL
szögpont alatt szerepel, akkor ez a szám a művelet-
nek megfelelő szögpont első attributumában található
érték. /14. ábra/

Megjegyzés: A műveletvégző rutinok a művelet eredményére
vonatkozó információt a RES/IRES/ változóba
helyezik el.

/NEG, AND, OR, NAND, NOR, XOR, IMPL, EKV,
1MIN, EGYSZ,
2RED, BDIFF, NANDS, NORS, ORANDZ, HERT,
3CHELY, SZINT, EGYMO,
4MUV1, MUV2,
UDUALIS, EGY, NULL, TBF, FUL/

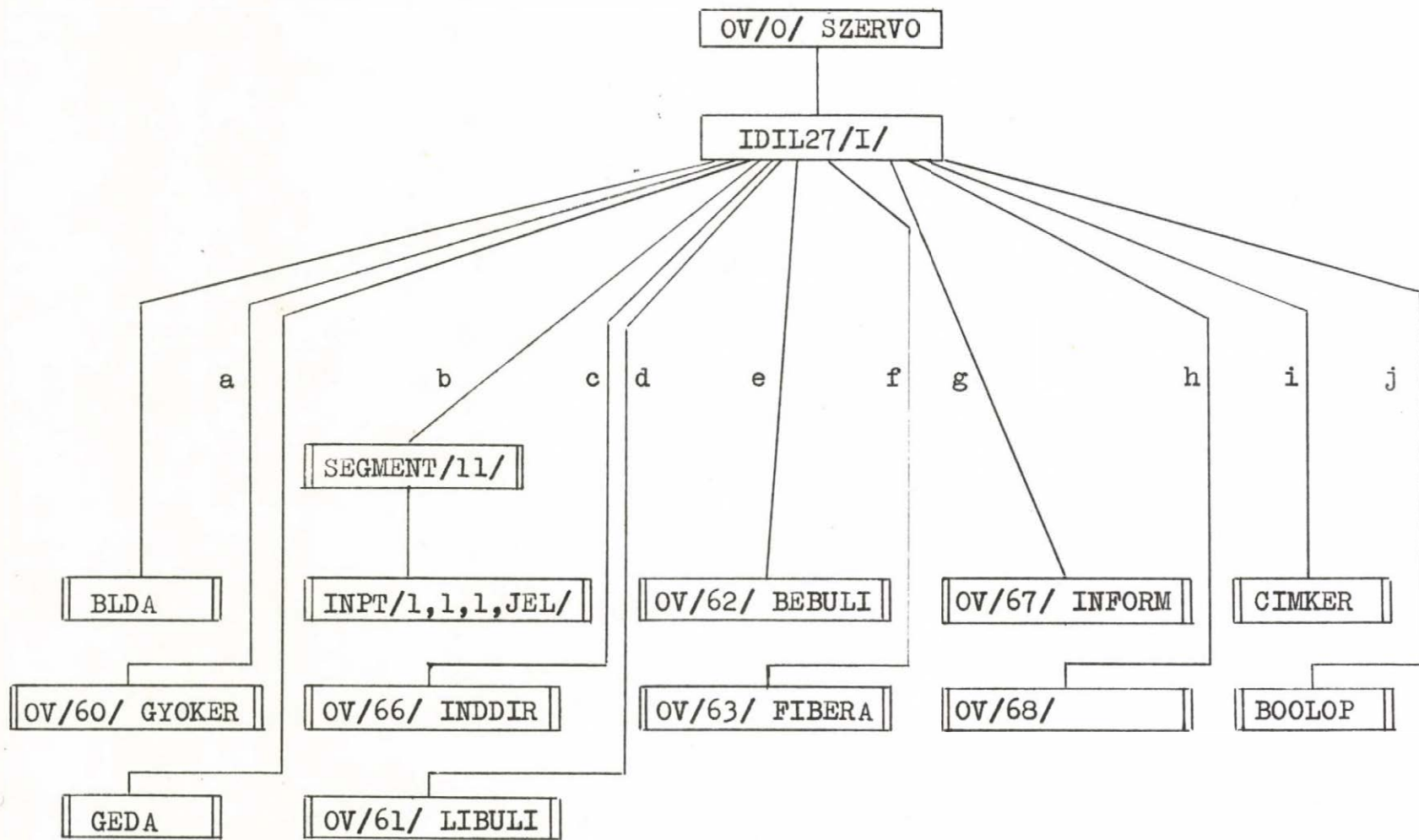
+

15.a ábra

/BOOLEFV/DEF/MUVJEL/UJMUV/,
X FVCSOP/NEG/EGY, NULL/, DUALIS/,
X FORMDEF,
X VALTLIST/EGY/EGY/=1, 1=//, NULL/NULL/=-1, -1=////,
X LEIRAS/FORMULA,
X NORMF/LINTER/AND, OR/,
X FATER /AND, OR/,
X FULTER/AND, OR//,
X BEOLVAS,
X MUVELET///

+

15.b ábra



16. ábra

39 NEG
42 AND
45 OR
48 NAND
51 NOR
54 XOR
57 IMPL
60 EKV
63 MIN
66 EGYSZ
69 RED
72 BDIFF
75 NANDS
78 NORS
81 ORANDZ
84 HERT
87 CHELY
90 SZINT
93 EGYMO
96 MUV1
99 MUV2
102 DUALIS
105 EGY
108 NULL
111 TBF
114 FUL

Művelet nevek

117 BOOLEFV
120 DEF
123 MUVJEL
126 UJMU
129 FVCSOP
132 FORMDEF
135 VALTLIST
138 LEIRAS
141 FORMULA
144 NORMF
147 LINTER
150 FATER
153 FULTER
156 BEOLVAS
159 MUVELET

Gyökér szögpont nevek

Az OVERLAY/66/ által aktivizált szegmensek működése

A vezérlőprogram és a szegmensek kapcsolatát a 17. ábra mutatja.

SUBROUTINE INDIT

- a. Elhelyezi a Boole műveletek belső neveit az OPER nevű közös adatmezőbe. /SINS/K/ LIDI72 rutin aktivizálása/
/3. táblázat/
- b. Elhelyezi a BFKR standard gyökerét alkotó szögpontok belső neveit a REPR nevű közös adatmezőbe. /SINS/K/ LIDI 72 rutin aktivizálása/ Nem kerülnek elhelyezésre azok a szögpontnevek, amelyek valamely művelet névvel azonosak.
/4. táblázat/
- c. Elhelyezi a CIMES közös adatmezőbe a CIMKER szubrutin aktivizálásával a standard gyökeret alkotó fa szögpontjainak címeit. /2. táblázat/

SUBROUTINE DIRECT

- a. A szubrutin végzi el az adatstruktúra kialakító műveletek vezérlőmezejének feltöltését, annak az OVERLAY lap-

nak a számával, amely a vezérlő mezőhöz rendelt művelet elvégzéséhez szükséges. A vezérlő mező és az első osztályba tartozó műveletek között a következő megfelelés érvényes:

művelet	a vezérlő mező művelethez rendelt elemének sorszáma
MUVELET	1
FORMULA	2
VALTLIST	3
FORMDEF	4
FVCSOP	5

Ha a listában az első három művelet valamelyike definiálva van, akkor a 61, ha a 4. művelet, akkor a 62, ha az 5. művelet, akkor a 63 kerül beírásra a megfelelő vezérlő mezőbe. Ha nincs definiálva művelet igény a listában, azt a rendszer úgy tekinti, hogy az 1-5 műveletek mindegyikét el kell végezni a listán.

- b. A szubrutin a művelet igény figyelembevételével megvizsgálja, hogy a definiált műveletek elvégezhető-e.

A hibajelzések a következők:

1. Ha az igény az összes művelet elvégzése és a MUVELET szögpont alól hiányzik a műveletleírás, akkor a hibaüzenet a következő:

*****MUEVELETLEIRAAS HIAANYAABAN A FELADAT NEM
VEEGREHAJTHATO.*****

2. Ha a MUVELET operáció elvégzése az igény, de hiányzik a MUVELET szögpont alól a műveletleírás, akkor a hibaüzenet:

*****MUEVELETVEEGZEES LEIRAAS NEELKUEL NEM
LEHETSEEGES.*****

3. Ha a BEOLVAS vagy a FORMULA szögpont terminális, akkor a hibaüzenet:

*****NINCS INFORMAACIO, AZ ERDOE NEM HELYEZHETOE
EL.*****

4. Ha a ϕ -Boole-függvények adminisztrációjához vagy a helyettesítéshez hiányzik a listából a szükséges információ, akkor a hibaüzenet:

*****A VAALTOZOLISTA HIAANYZIK A FIBOOLE FUEGGVEENYEK
ADMINISZTRAACIOJAAHOZ VAGY A HELYETTESITEESHEZ.*****

5. Ha MUVELET és FORMULA operációt nem irtunk elő, továbbá a BEOLVAS és a FORMULA szögpont terminális, akkor a hibaüzenet:

*******NINCS INFORMACIO A VALTLIST, FORMDEF, FVCSOP
MUEVELET ELVEEGZEESEEHEZ.*******

A felsorolt esetekben BFKR befejezi működését.

6. Ha a MUVELET szögpon t alatt van műveletleírás, de a MUVELET operációt nem irtuk elő, akkor a hibaüzenet:

*******NEM KEERI A LEIRT MUEVELET ELVEEGZEESEET.*******

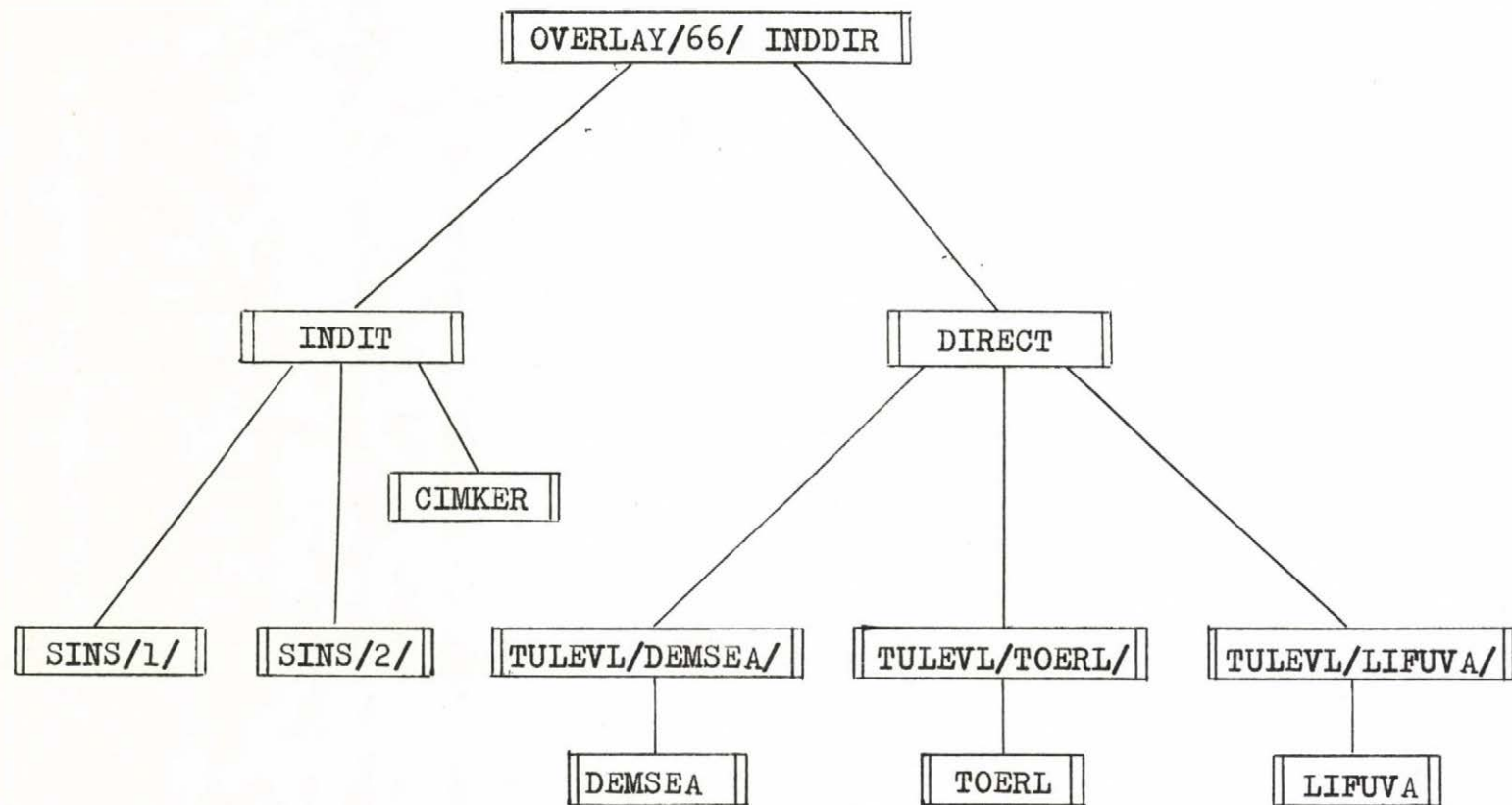
7. Ha a FORMULA operációt előirtuk és a BEOLVAS szögpon t terminális, de a FORMULA szögpon t nem, akkor a hibaüzenet:

*******AZ ERDOE ELHELYEZEESE NEM VEEGREHAJTHATO.*******

8. Ha a BEOLVAS szögpon t alatt van kiegészítő formátumban megadott információ, de sem FORMULA, sem VALTLIST műveleteket nem irtunk elő, akkor a hibaüzenet:

*******NEM KEERI A KIEGEESZITOE FORMAATUMBAN MEGADOTT
INFORMACIO FELDOLGOZAASAAT.*******

A 6-8 üzenetek kiírása után BFKR végrehajtása folytatódik.



17. ábra

/CIMES/ adatmező felosztása

Azonosító A szögpont külső neve, amelynek fejcime az
azonosítóval meghatározott helyre kerül

1	BFVCIM	BOOLEF	
2	DEFCIM	DEF	
3	LIRCIM	LEIRAS	
4	MJCIM	MUVJEL	
5	FVCSCM	FVCSOP	
6	FDEFM	FORMDEF	
7	VACIM	VALTLIST	
8	FOCIM	FORMULA	
9	NFCIM	NORMF	
10	BEOLVC	BEOLVAS	
11	MUVCIM	MUVELET	
12	LICIM	LINTER	
13	FTCIM	FATER	
14	FULCIM	FILTER	
15	NEGCIM	NEG	
16	DUALCM	DUALIS	
17	LANDC	AND	/LINTER külső nevü szögpont alatt/
18	LORC	OR	/LINTER külső nevü szögpont alatt/
19	FANDC	AND	/FATER külső nevü szögpont alatt/
20	FORC	OR	/FATER külső nevü szögpont alatt/
21	FUANDC	AND	/FILTER külső nevü szögpont alatt/
22	FUORC	OR	/FILTER külső nevü szögpont alatt/
23	UJMCIM	UJMUJ	
24	PROCIM	PROC	

/OPER/ közös adatmező felosztása

Azonosító A külső név, amelynek belső neve az azonosítóval meghatározott helyre kerül

1	NEG	NEG
2	AND	AND
3	OR	OR
4	NAND	NAND
5	NOR	NOR
6	XOR	XOR
7	IMPL	IMPL
8	EKV	EKV
9	MIN	MIN
10	EGYSZ	EGYSZ
11	RED	RED
12	BDIFF	BDIFF
13	NANDS	NANDS
14	NORS	NORS
15	ORANDZ	ORANDZ
16	HERT	HERT
17	CHELY	CHELY
18	SZINT	SZINT
19	EGYMO	EGYMO
20	DUAL	DUALIS
21	EGY	EGY
22	NULL	NULL
23	TBF	TBF

/REPR/ közös adatmező felosztása

Azonosító A szögpont külső neve, amelynek belső neve
az azonosítóval jelölt helyre kerül

1	BFV	BOOLEFV
2	DEF	DEF
3	LEIR	LEIRAS
4	MJEL	MUVJEL
5	FVCS	FVCSOP
6	FDEF	FORMDEF
7	VA	VALTLIST
8	FO	FORMULA
9	NF	NORMF
10	BEOL	BEOLVAS
11	MUV	MUVELET
12	LI	LINTER
13	FT	FATER
14	FUL	FILTER
15	UJM	UJMUJ
16	PROC	PROC

4. táblázat

Az adatstruktúra kialakító műveleteket végrehajtó rutinok

A műveleteket végrehajtó OVERLAY lapok vezérlő programjainak és az általuk aktivizált szegmenseknek a kapcsolatát a 19-21. ábrák mutatják.

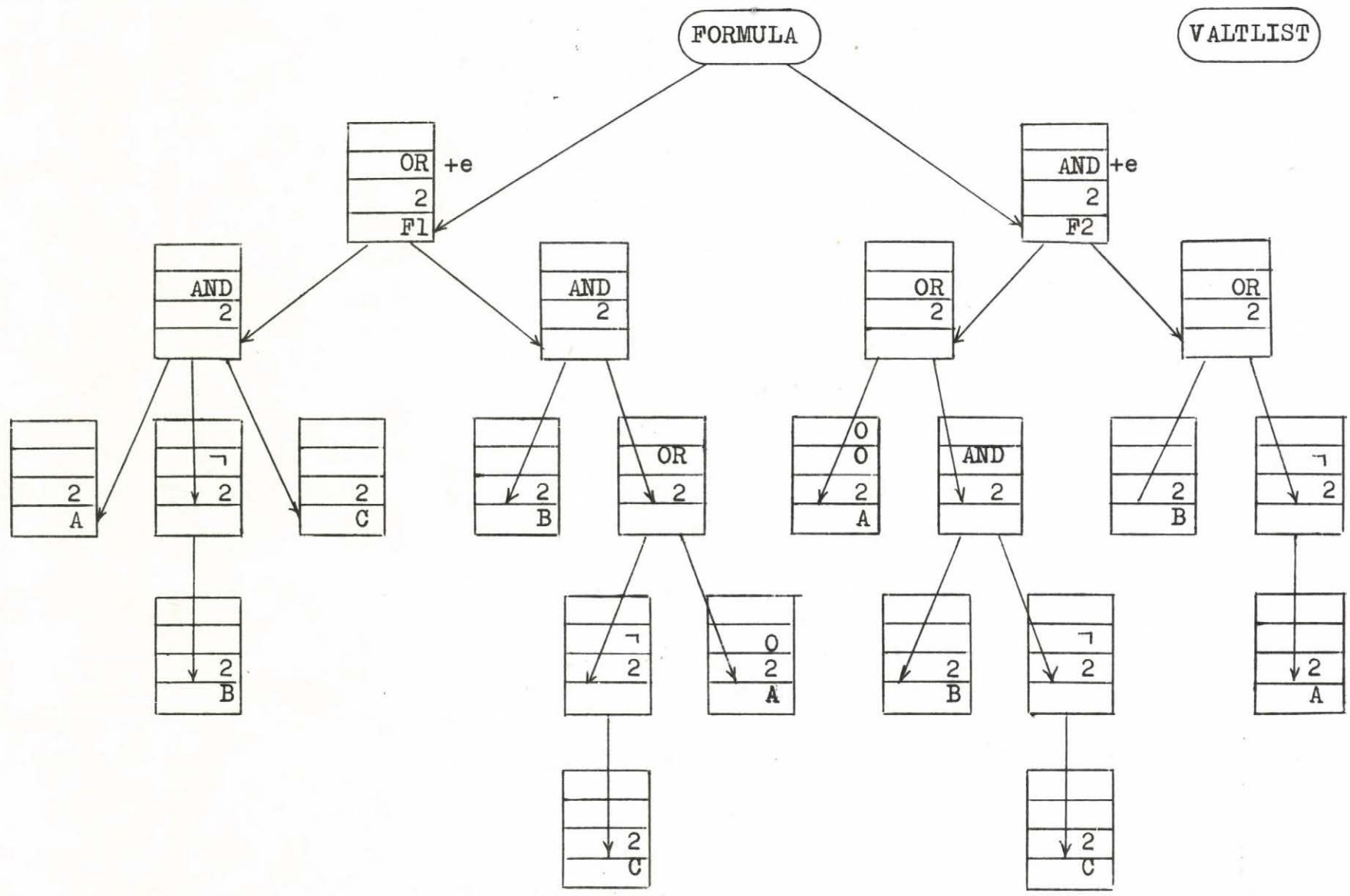
SUBROUTINE VLISTI/IAMUV,HIBUCI/

A /MUNK/ közös adatmezőben megadott KIV értékétől függően a következő feladatokat hajtja végre.

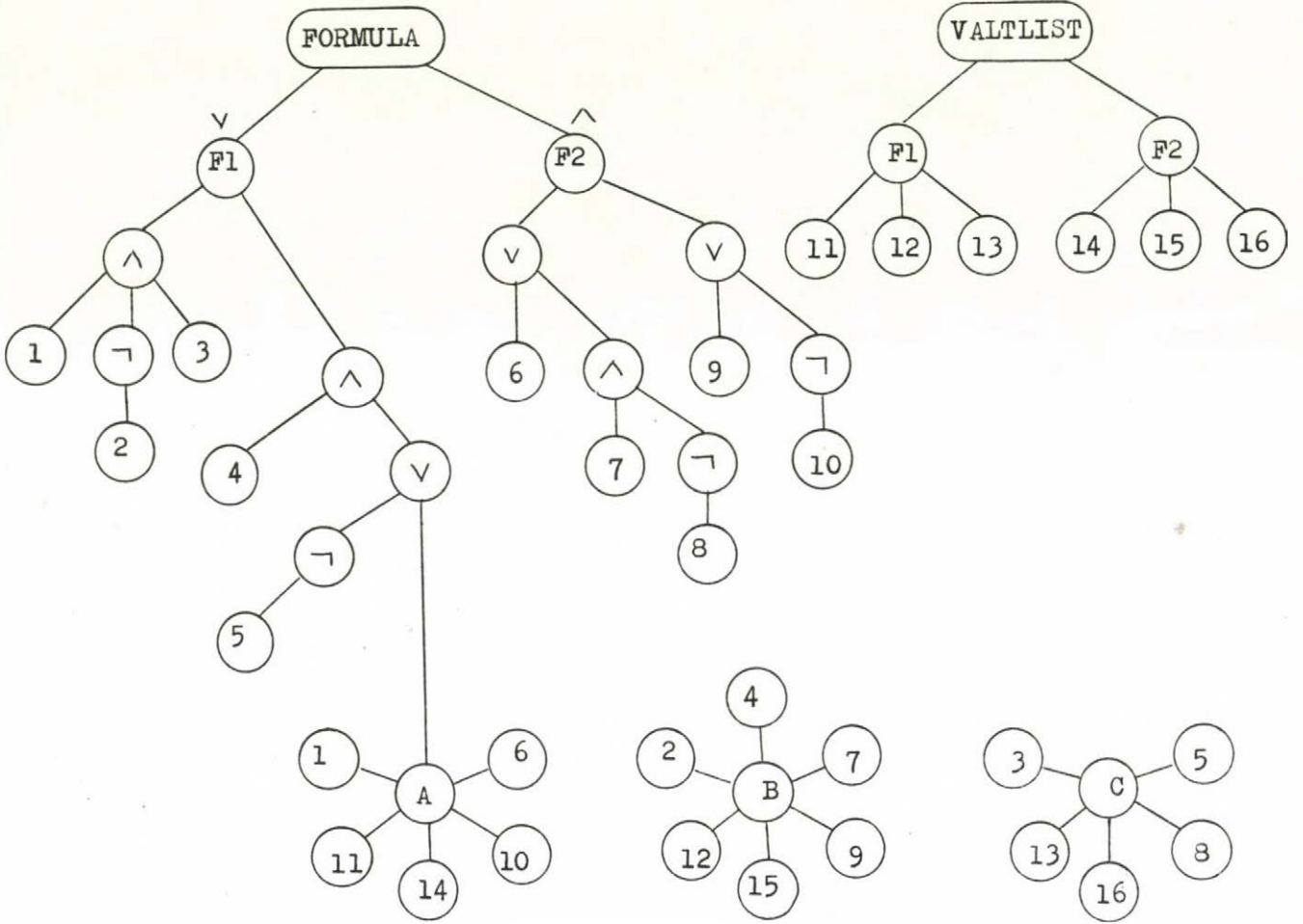
1. Ha KIV=1, akkor a MUVELET szögpontra alatt formulaként megadott műveletleírás szintaktikai vizsgálatát végzi el. /SINTER szubrutin aktivizálása/
2. Ha KIV=2, akkor
 - elvégzi a BEOLVAS szögpontra alatt megadott formula halmaz szintaktikai vizsgálatát. /SINTER szubrutin/
 - ha a szintaktikai vizsgálat eredménye jó, akkor az egyes formulák formulagráfjait elhelyezi a FORMULA szögpontra alá. /REPOL, EXPREB vagy ASSZIM szubrutinok egymásutáni aktivizálása/ Egyébként a vezérlés visszatér a főprogramba.

3. Ha KIV=3, akkor végrehajtja a 2. pontnál leírt műveleteket. Ezenkívül a FORMULA szögpontra alá elhelyezett Boole-függvények változólistáit elhelyezi a VALTLIST szögpontra alá. /FLIST szubrutin/

Egy szögpontra vonja össze a FORMULA szögpontra alá elhelyezett formulagráfok többszörösen előforduló azonos nevű szögpontjait. A létrehozott gráfot pl. az $F1 = A \times \neg B \times C + B / \neg C + A /$ és $F2 = /A + B \times \neg C // B + \neg A /$ függvények esetén a 18. a, b ábra mutatja. Az a ábra a 2. pontnál, a b a 3. pontnál leírt műveletek végrehajtása utáni állapotot mutatja. A b ábrán rövidítve tüntettük fel a szögpontok attribútumait. A \bigcirc -be üres nevű szögpontra esetén az attribútumban szereplő műveleti jel szimbóluma került beírásra. $\bigcirc 1$, $\bigcirc 2$, ... szimbólumok az él folytatását jelölik.



18.a ábra



18.b ábra

A KIV értékétől függetlenül törli a BEOLVAS szögpontra alatt megadott információt és tömöríti a listát.

Ha IAMUV bemenő paraméter értéke 1 ill. 0, akkor a szubrutin képes ill. nem képes nem kommutatív műveleteket tartalmazó formulákat kezelni.

A VLISTI szubrutinnal aktivizált rutinok működése

SUBROUTINE SINTER/IFL,KSI,HIBUCI/

Az IFL bemenő paraméterrel megadott nevű szögpontra alatt elhelyezett, kiegészítő formátumu információ szintaktikai vizsgálatát végzi el követési táblázat segítségével. A követési táblázat kitöltése az ADATOK nevű szubrutin feladata. A szintaktikai vizsgálat egyidőben történik a kiegészítő formátumban megadott információ átalakítása is.

Az átalakítás lényege, hogy a Boole formula leírásában műveleti jelként használható elválasztó jeleket a szubrutin átirja a művelet külső nevéhez rendelt belső névvel.

A KSI kimenő paraméter értéke 0 ill. 1 attól függően, hogy a rutin nem talált ill. talált kiegészítő formátumban megadott információt.

A HIBUCI kimenő paraméter értéke 1, ha a formulában volt szintaktikai hiba, egyébként 0.

A szubrutin hibajelzései a következők:

1. Ha az IFL szögpont terminális, kiírásra kerül a

NINCS VIZSGAALANDO FORMULAHALMAZ

szöveg.

2. Ha valamely formula nem névvel kezdődik, akkor kiírásra kerül a következő üzenet:

*****A NEEV NEM FUEGGVEENYNEEV*****

A NEEV INDEXE A LISTAABAN: IND

A NEEV BELSOE NEVE:..... L/IND/

A FUEGGVEENY SORSZAAMA:... KS

3. Ha a függvéynév után nem következik egyenlőségjel, kiírásra kerül a következő üzenet:

*****A FUEGGVEENYNEVET NEM KOEVETI EGYENLOESEEGJEL*****

A KARAKTER INDEXE A LISTAABAN: IND

A KARAKTER BELSOE NEVE: L/IND/

A FUEGGVEENY SORSZAAMA: KS

4. Ha a formula neve a FORMULA szögpont alá korábban elhelyezettel azonos, akkor kiírásra kerül a következő üzenet:

*****A FUEGGVEENY MAAR SZEREPEL A VAALTOZOLISTA SZOEGPONT
ALATT*****

A FUEGGVEENY BELSOE NEVE: FNLV/IXY/

5. Ha a követési táblázat szerint két karakter egymást nem követheti /karakternek nevezzük az egységes belső ábrázolás miatt a változó nevet is/, akkor kiírásra kerül a következő üzenet:

*****A KARAKTER AZ OET MEGELOEZOE UTAAN NEM KOEVETKEZ
HET*****

A KARAKTER INDEXE: IND

A KARAKTER BELSOE NEVE: L/IND/

A FUEGGVEENY SORSZAAMA: KS

6. Ha két olyan karakter követi egymást, amely a karakterpárt követőt generálja, és az nem a generált, akkor kiírásra kerül a következő üzenet:

*****A KARAKTERT KOEVETOE KARAKTER ILLEGAALIS*****

A VIZSGAALT KARAKTER INDEXE: IND

A VIZSGAALT KARAKTER BELSOE NEVE: L/IND/

A FUEGGVEENY SORSZAAMA: KS

7. Ha két olyan karakter követi egymást, amely a karakterpárt megelőzőt generálja, és ez nem a generált, akkor kiírásra kerül a következő üzenet:

*****A KARAKTERT MEGELOEZOE, MAASODIK KARAKTER ILLEGAALIS*****

A KARAKTER INDEXE: IND

A KARAKTER BELSOE NEVE: L/IND/

A FUEGGVEENY SORSZAAMA: KS

8. Ha a Boole-függvény leírásában nem megengedett elválasztójel van, akkor kiírásra kerül a következő üzenet:

~~*****~~ILLEGAALIS ELVAALSZTOOJEL A LISTAABAN~~*****~~

A KARAKTER INDEXE A LISTAABAN: IND

A FUEGGVEENY SORSZAAMA:..... KS

9. Ha a vizsgálat során egy /-hez értünk és az addig talált zárójelpárok mind teljeseek, akkor kiírásra kerül a következő üzenet:

~~*****~~FELESLEGES VEEGZAAROOJEL~~*****~~

A KARAKTER INDEXE A LISTAABAN: IND

A FUEGGVEENY SORSZAAMA:..... KS

10. Ha a formula vizsgálat végén még marad kezdőzárójel, amelyhez nem tartozik végzárójel, akkor kiírásra kerül a következő üzenet:

~~*****~~BE NEM FEJEZETT ZAAROJELPAAROK A FUEGGVEENYBEN~~*****~~

A ZAAROJELPAAROK SZAAMA: IZAR

A FUEGGVEENY SORSZAAMA:.. KS

A 2-10. sorszámú hibajelzések a HIBUCI kimenő paraméter értékét 0-ról 1-re változtatják.

Üzenetek a szintaktikai vizsgálat elvégzése után:

1. Ha a rutin a feldolgozás során talált 0 nevű szögpon-
tot, akkor az attributumok sorozatán végrehajtotta a feladatot,
és kiírásra kerül a következő üzenet:

A MOST VIZSGAALT FORMULAHALMAZOK SZAAMA: NSINT A KORAABBAN
VIZSGAALTAKEE: NCI

Ha az IFL nevű szögpon-
t alatt lehet nem kiegészítő for-
mátumu információt tartalmazó szögpon-
t is, akkor a rutin
annak minőségét nem vizsgálja, hanem a korábban vizsgált
formulahalmazokhoz számolja.

2. Ha a rutin a feldolgozás során nem talált 0 nevű szög-
pontot, akkor kiírásra kerül

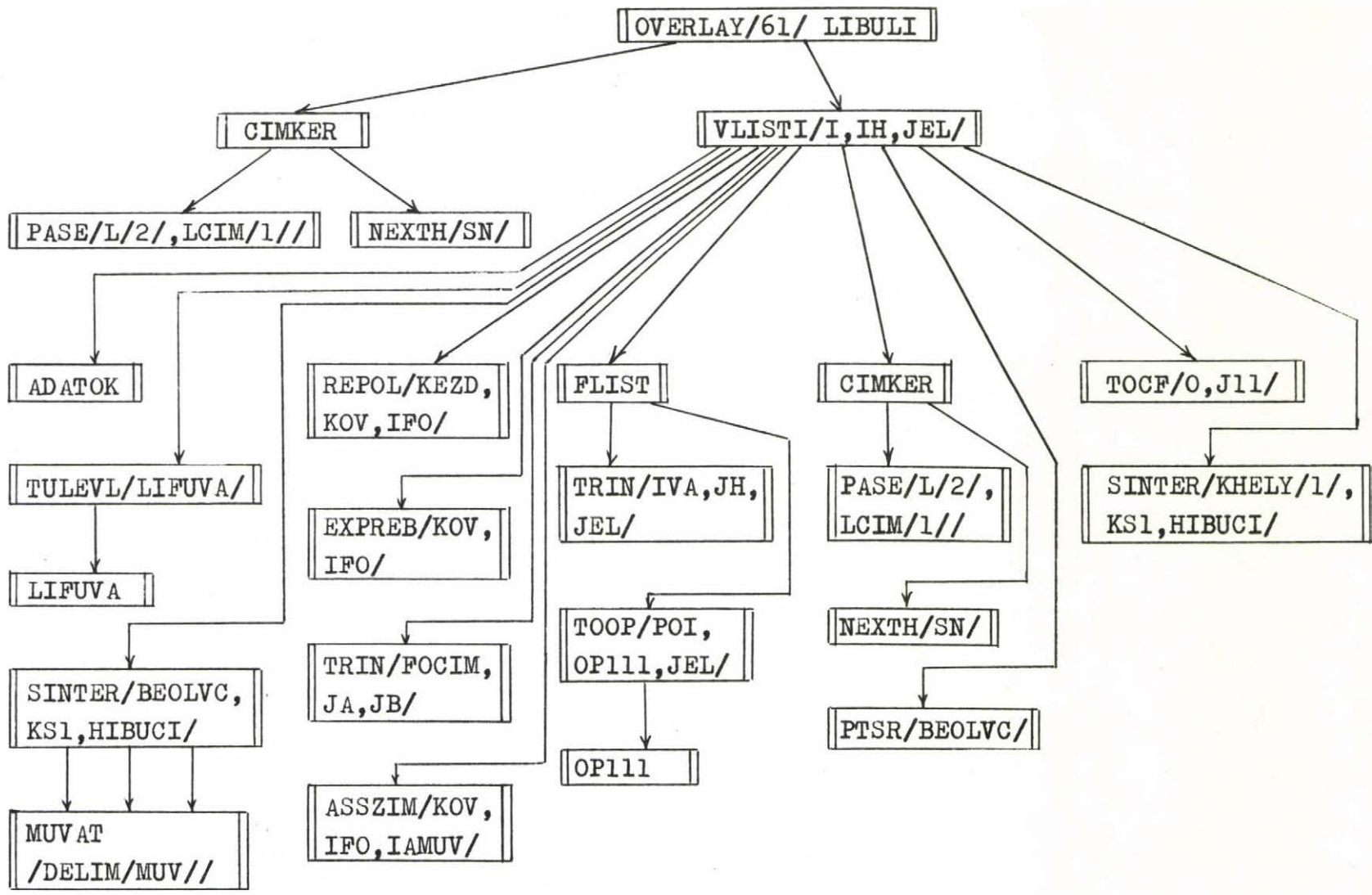
~~*****~~A SZINTAKTIKAI VIZSGAALAT EGY KORAABBI ALKALOMMAL
MEGTOERTEENT VAGY NEM NULLA NEVUE FEJ A VIZSGAALT
SZOEGPONT ALATT~~*****~~

szöveg.

3. Ha minden vizsgált szögpon-
t 0 nevű volt, akkor kiírásra
kerül

A FORMULARENDSZER TELJES SZINTAKTIKAI VIZSGAALATA MOST
TOERTEENT

szöveg.



19. ábra

SUBROUTINE REPOL/K1,K2,K/

A szubrutin elkészíti a paraméterátadó közös területbe egy szintaktikailag helyes formula fordított lengyel formáját, ahol a K1, K2 bemenő paraméterek megadják a formula első illetve utolsó elemének címét. A K kimenő paraméter a fordított lengyel formulát tartalmazó verem aktuális pointere. Ha a formula a verem maximális méretét /1000 félszó/ túlhaladja, akkor kiírásra kerül a

*******A LENGYEL FORMULA NEM ELHELYEZHETOE.*******

A FORMULA BELSOE NEVE: n

üzenet, és a programrendszer működését befejezi.

SUBROUTINE EXPREB/K,IFO/

A szubrutin a paraméterátadó közös területben elhelyezett fordított lengyel formulából elkészíti a zárójeles formulát a paraméterátadó közös terület /LIDI 72 listakezelő rendszer által/ előirt részében. A szubrutin formális paraméterei:

K - a formula első elemének indexe

IFO - a fordított lengyel formulát tartalmazó verem pointere

A zárójeles formulában minden szögpont leírásában 2 attributum szerepel.

A formulát ábrázoló fa gyökérszögpontjának leírásában a szögpont neve a formula neve, ehhez járul a 19. biten az 1, ha a formula neve előtt a külső leírásban negáció jele szerepelt, továbbá az első attributum tartalmazza a kifejezés legalacsonyabb precedenciájú műveleti jelét és a vizsgálati jelet. E jel szerepe, hogy megkülönböztesse a változólistával rendelkező és nem rendelkező formulákat.

A műveleti jel szögpontok leírásában a név üres, az első attributum tartalmazza a műveleti jelet.

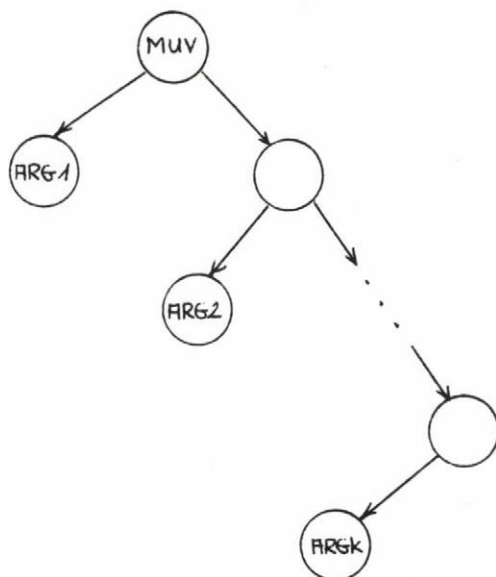
A változó szögpontok leírásában a két attributum üres, a szögpont neve a változó nevével egyezik meg. A szubrutin csak olyan Boole formulák feldolgozására képes, amelyek nem tartalmaznak nem kommutatív Boole műveleteket.

SUBROUTINE ASSZIM/K,IFO,IAMUV/


A szubrutin előállítja olyan formula fordított lengyel formulájából annak zárójeles formuláját a paraméterátadó közös terület /LIDI 72-ben/ előírt részében, amely tartalmaz nem kommutatív műveletet is.

Működése csaknem azonos az EXPREB rutinével, ezért csak az eltéréseket közöljük.

Ha a lengyel formulában nem kommutatív műveletet talál, akkor elhelyezi a 20. ábrán látható gráfnak megfelelően a zárójeles formulát.



20. ábra

Az ábrán  -vel jelölt szögpontok üres nevek és attributum nélküliek.

SUBROUTINE FLIST

A szubrutin a FORMULA szögpontra alatt található azon formulák változólistáit készíti el, amelyeknek gyökere az első attribútumában tartalmaz vizsgálati jelet.

Egy adott formula változólistájának elkészítése két lépésben történik.

- a. A változólista gyökerének elhelyezése a VALTLIST szögpontra alá. /18.b ábra/
- b. A gyökér alá a változók elhelyezése.

Ennek során a program az adott formula formulagráfját bejárva /18.a, b ábra/, annak terminális elemeivel végez műveleteket /OP111 rutin/.

Ha a terminális elemmel azonos nevű már szerepel a VALTLIST szögpontra alatt, valamely nem az adott függvény változójaként, akkor a vizsgált szögpontra töröljük a listából és a hozzá vezető életra a vele azonos nevű szögpontra irányítjuk. Továbbá ezt a szögpontra összekapcsoljuk az adott függvény változólistájának gyökérszögpontraival is.

Ha a terminális elemmel azonos nevű az adott függvény változójaként szerepel a VALTLIST alatt, akkor csak az előbbi művelet első része kerül végrehajtásra.

Ha a változólisták bejárása során nem találtunk a terminális elemmel azonos nevűt, akkor a szögpontot bekapcsoljuk az adott függvény változólistájának gyökeréhez.

Ha a függvény változólistája elkészült, a vizsgálati jelet a formulagráf gyökeréből töröljük.

SUBROUTINE SIEQUS

A szubrutin a /MUNK/ közös adatmezőben elhelyezi a NET szubrutin működéséhez szükséges kezdőértékeket és aktivizálja azt.

SUBROUTINE NET/JEL/

A szubrutin elvégzi a formulahalmazban lehetséges összes helyettesítést. A végrehajtás lépései a következők:

- a. A paraméterátadó közös területben a változólisták gyökérszögpontjaira vonatkozó információt állít elő. A gyökérszögpontok neveit az FNLV, címeit az FSLV tömbök-

be párhuzamosan helyezi el. A feladat végrehajtásához a NET aktivizálja a TULEVL bejáró rutint, amely a LIFUVA external rutin segítségével végzi el a tömbök feltöltését.

Ha a változólisták száma meghaladja a verem maximális méretét /100/, akkor a NET a behelyettesítést /FORMDEF művelet/ nem hajtja végre.

- b. A rutin elhelyezi az FSLF tömbbe a formulagráfok gyökérszögpontjainak címeit, úgy hogy az FNLV/I/ nevű formula címe kerül az FSLF/I/-be.
- c. Kigyűjti a FORMDEF szögpont alatt szereplő, behelyettesítéssel nyert összetett nevű függvények változólistáinak gyökérszögpontjaira vonatkozó információkat a paraméterátadó közös területbe; a CONAFD tömbbe a változólisták gyökereinek neveit /formula nevek/ és a CONSF D tömbbe a gyökerek címeit. A feladat végrehajtása a TULEVL bejáró rutin aktivizálásával történik, amely az egyes szögpontokban a feladatot a LINEWL rutin segítségével hajtja végre.
- d. A behelyettesítés előkészítése. A rutin megvizsgálja, hogy a változó neve előfordul-e formula névként is.

Ha igen, megvizsgálja, hogy a formula a korábbiakban behelyettesítésre került-e már. Ha nem, akkor a formula gyökerének címét elhelyezi a változót tartalmazó szögpont első attributumába és a FORMULA szögpont alóli törléshez szükséges munkamezőbe. /TULEVL bejáró rutin függvényenkénti aktivizálása a VACOFU external rutinnal./

- e. Megvizsgálja, hogy a behelyettesítés sorozata nem vezethet-e implicit függvény keletkezéséhez. Ha a behelyettesítések valamely sorozatában egy olyan formula fordul elő, amely a sorozatban már korábban is előfordult, akkor kiírja a

~~*****~~BEHELYETTESITEES IMPLICIT FUEGGVEENY KIALAKULAASAAHOZ
VEZETETT.~~*****~~

üzenetet és a behelyettesítés sorozatban szereplő formulák formulagráfjainak gyökércímeit.

A feladatot NET az IMPLIC rutin segítségével hajtja végre, amelynek kimenő paramétere tájékoztatja a rutint a hiba bekövetkezéséről. Hiba esetén az attributumokban elhelyezett címek törlődnek és a behelyettesítés művelete /FORMDEF/ nem hajtódik végre.

f. A TOOPJ bejáró rutint aktivizálja, amely a JOINOP external rutin segítségével elvégzi a behelyettesítéseket. A behelyettesítés azt jelenti, hogy a szögpontra mutató pointert kicseréli a szögpont első attributumában tárolt címre. A behelyettesített függvény nevét egy veremben tárolja, amelynek legalsó eleme azon függvény belső nevének -l-szerese, amelybe a behelyettesítés történt. A formula gyökérszögpontjának címét elhelyezi az MFSLF tömb azon elemébe, amelynek indexe megegyezik a behelyettesített függvény nevét tartalmazó FNLV tömbelem indexével.

Ha a behelyettesített függvények vermeinek tárolására biztosított tömb betelt, kiírásra kerül a

~~*****~~TULCSORDULAAS CONAME TOEMBBEN.~~*****~~

üzenet és a program végrehajtása megszakad.

- g. A FORMULA szögpontból a behelyettesített függvények gyökereihez vezető élek törlése /BRRW LIDI 72 rutin/.
- h. Beírja a paraméterátadó közös terület SYCONA ill. SYCOSS tömbjeibe az összetett nevek ill. az összetett nevek komponenseit tartalmazó szögpontok címeit. A legkisebb értékű összetett nevet a LIDI TECM rendszerváltozójába írjuk.

Ha az összetett nevek száma 200-nál több, akkor kiírásra kerül a

~~*****~~TULCSORDULAAS SYCONA EES SYCOSS TOEMBOEK BEN.~~*****~~

üzenetet és a program végrehajtása befejeződik.

A műveletet a NET a MYCNOU rutin segítségével hajtja végre.

- i. A FORMDEF alatt szereplő összetett nevű szögpontok listájában a neveket átírja azok első komponenseivel.
/FICOMP rutin/
- j. A NEWL1 szubrutin segítségével létrehozza a FORMDEF alatt a behelyettesítéssel nyert új függvények változólistáit ill. a régi változólistákhoz új neveket generál.

A következő esetek fordulhatnak elő:

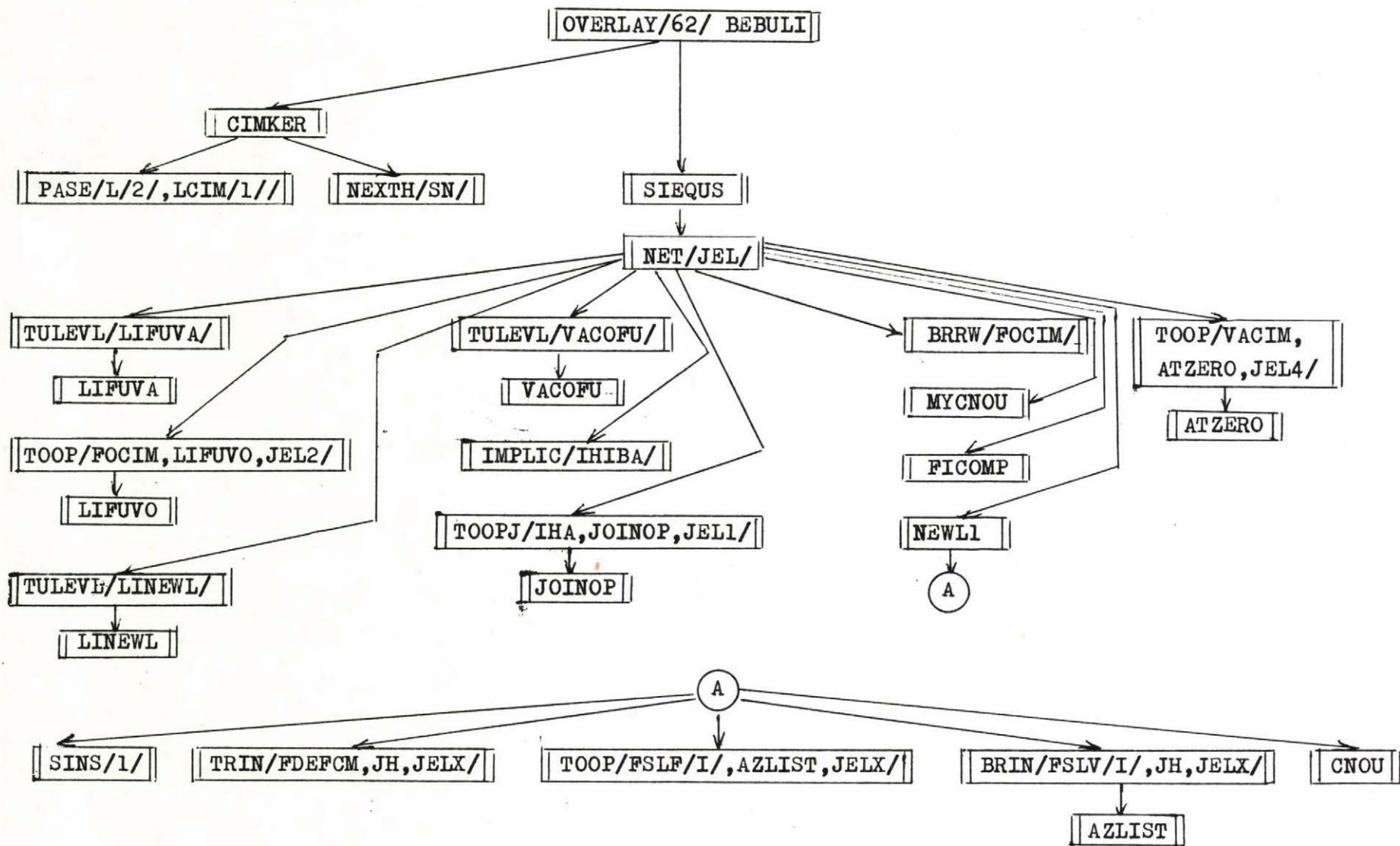
1. A behelyettesítés olyan F nevű függvénybe történt, amelybe korábbi alkalommal még nem történt helyettesítés. Ekkor a változólista F.FUL névvel kerül elhelyezésre a FORMDEF alá.
2. A behelyettesítés olyan F nevű függvénybe történik, amelyhez a FORMDEF alatt az F.FUL nevű változólista található. Ebben az esetben létrejön F.FUL névvel az

új változólista és a régi változólista neve $F.G_1 \dots G_k$ lesz, ahol a G_i -k $1 \leq i \leq k$ az F -be helyettesített új függvények nevei.

3. A behelyettesítés olyan F nevű függvénybe történik, amelyhez található az $F.FUL$ és az $F.G_1.G_2 \dots G_k$ nevű változólista is, azaz már korábban legalább kétszer történt helyettesítés. Ebben az esetben a 2. pontban leírt változólistán kívül létrejön az $F.G_1^i.G_2^i \dots G_k^i$. $G_1 \dots G_k$ összetett nevű változólista is, ahol $G_i^i / 1 \leq i \leq k$.

4. Ha az F függvénybe behelyettesítés történik, de ő maga is behelyettesítésre kerül F_1 -be, akkor csak azoknak a változólistáknak a neve változik, amelyek F -fel kezdődnek és új változólista nem kerül elhelyezésre.

k. A szögpontok első attribútumaiból a FORMULA szögpont alatti fá bejárásával törli az első attribútumokba elhelyezett munkainformációt.



20. ábra

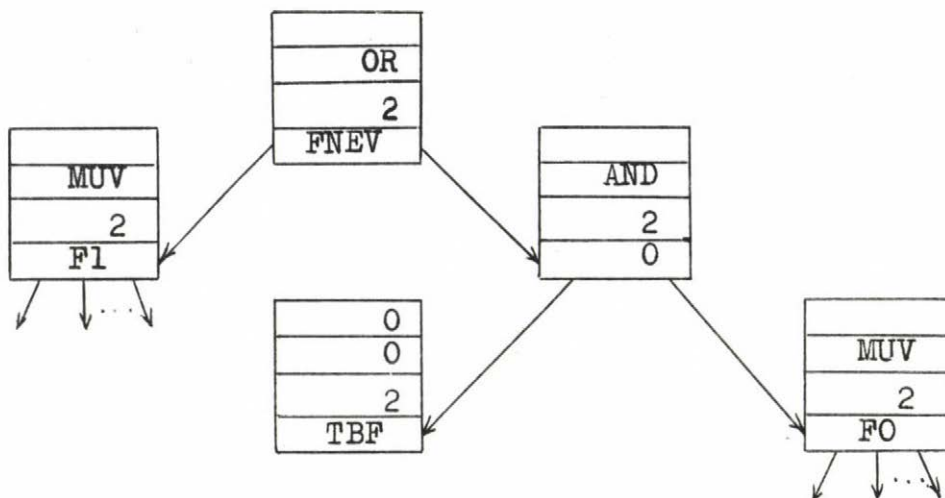
SUBROUTINE FIBULI

A szubrutin kikeresi a listából a Φ -Boole-függvényeket és a rájuk vonatkozó információkat /ha ez még nem történt meg/ elhelyezi a FVCSOP szögpont alá.

Az elhelyezésre kerülő információ:

0
FO
0
F1
4
FNEV

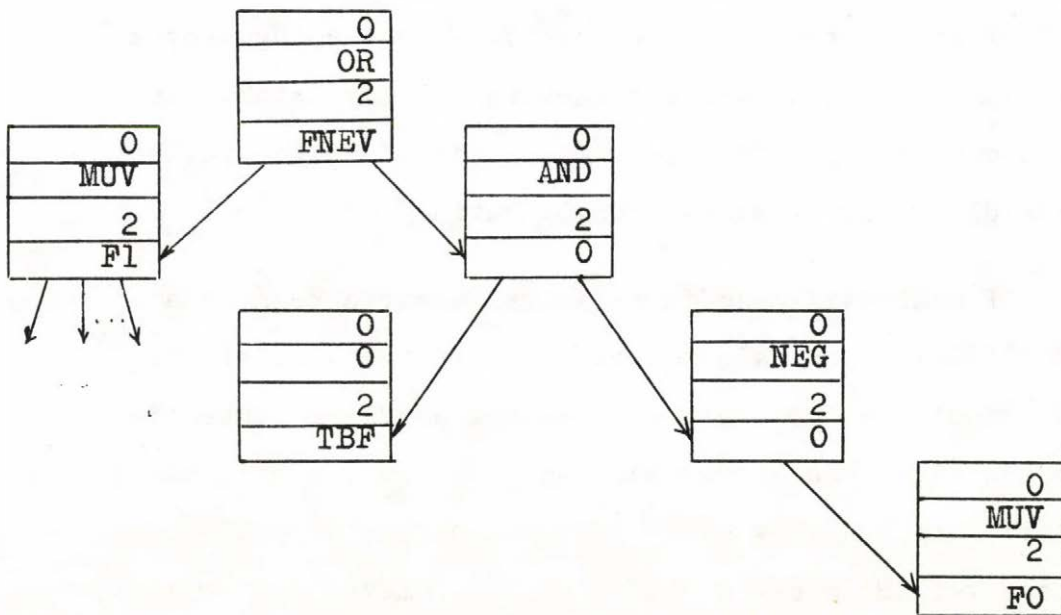
, ha a Φ -Boole-függvényt leíró gráf



alaku, és

0
0
FO
F1
4
FNEV

, ha a ϕ -Boole-függvényt leíró gráf



alaku.

Valamely függvényt akkor tekint a rutin Φ -Boole-függvénynek, ha változólistáján a TBF változó szerepel. Minden olyan függvénymegadást hibásnak tekint, amelynek a változói között a TBF szerepel, de a formulagráfjaik a fent megadottaktól különböznek.

A feladat végrehajtásának lépései a következők:

- a. A FIBULI által aktivizált LEOP szubrutin bejárja a FVCSOP szögpontra elhelyezett, Φ -Boole-függvényekre vonatkozó információkat tartalmazó szögpontokat és azok neveit kigyűjti a paraméterátadó közös területben kijelölt FULICS tömbbe. /FUNACS rutin/
- b. A LEOP szubrutin második aktivizálásakor a bejárás a VALTLIST szögpontra történik a változók szintjén. Az aktuális szögpontra a műveletet a FUNAME szubrutin hajtja végre. Ha az aktuális szögpontra TBF, a bejárás alatt álló függvény nevét összehasonlítja a FULICS tömb elemeivel. Ha a név a FULICS egyik elemével nem egyezik, a nevet beírja a fel nem dolgozott Φ -Boole-függvények listájába a paraméterátadó közös terület FIBUNA nevű tömbjébe.

c. Elkészíti a fel nem dolgozott Φ -Boole-függvények formulagráfjai gyökércímeinek listáját. /FIFOIN tömb/ a TULEVL bejáró rutin és a szögpontokban aktivizált LIFUVO műveletvégző rutin segítségével.

A b. és c. végrehajtása után a FULICS és a FIFOIN tömb azonos indexű elemei ugyanarra a függvényre vonatkozó információkat tartalmaznak.

d. Megvizsgálja, hogy a fel nem dolgozott függvények listáján szerepel-e olyan formulának a neve, amely a feldolgozás előtt behelyettesítésre került. Ha van ilyen függvény, akkor kiíródik a

~~*****~~ K. FUEGGVEENYNEK NINCSEN OENAALLO FORMULAGRAAFJA.~~*****~~

A FUEGGVEENY BELSOE NEVE: N1

üzenet.

Ahol k a név indexe a FULICS tömbben. Ha van ilyen függvény, akkor a vizsgálatot végző FIHIBA rutin kimenő paraméterét 0-ról 1-re változtatja és a rutin végrehajtása után a FIBULI a program futását felfüggeszti.

e. A paraméterátadó közös területben elkészíti a FVCSOP alá kerülő szögpontok zárójeles leírását. Ha kiderül, hogy a TBF változónevet tartalmazó formulagráf nem Φ -Boole-függvény, a felfedezett hibától függően a következő hibüzenetek valamelyike kerül kiírásra.

1. Ha a változólista alapján megállapítható f -ről, hogy Φ -Boole-függvény, és a formulagráf gyökérszögpontjából hiányzik az OR, akkor a hibaüzenet:

~~*****~~A FUEGGVEENY NEM DISZJUNKCIO~~*****~~

A FUEGGVEENY INDEXE A LISTAABAN: n

2. Ha a gyökérben szereplő OR vagy a második szinten /a gyökér szintszámát 1-nek véve/ szereplő AND 2-nél több pontert tartalmaz, akkor a hibaüzenet:

~~*****~~HA LX=0 A FUEGGVEENY ARGUMENTUMAINAK, HA LX=1

A KONJUNKCIO ARGUMENTUMAINAK SZAAMA TOEBB 2-NEEL~~*****~~

LX= n_1 A FUEGGVEENY INDEXE A LISTAABAN: n

3. Ha a gyökér vagy az AND-et tartalmazó szögpont alá bekapcsolt valamelyik szögpont hibás, akkor a hibaüzenet:

~~*****~~HA LX=0, A DISZJUNKCIO EGYIK ARGUMENTUMA HIBAAS,

LX=1 A KONJUNKCIOEE~~*****~~

LX= n_1 A FUEGGVEENY INDEXE A LISTAABAN: n

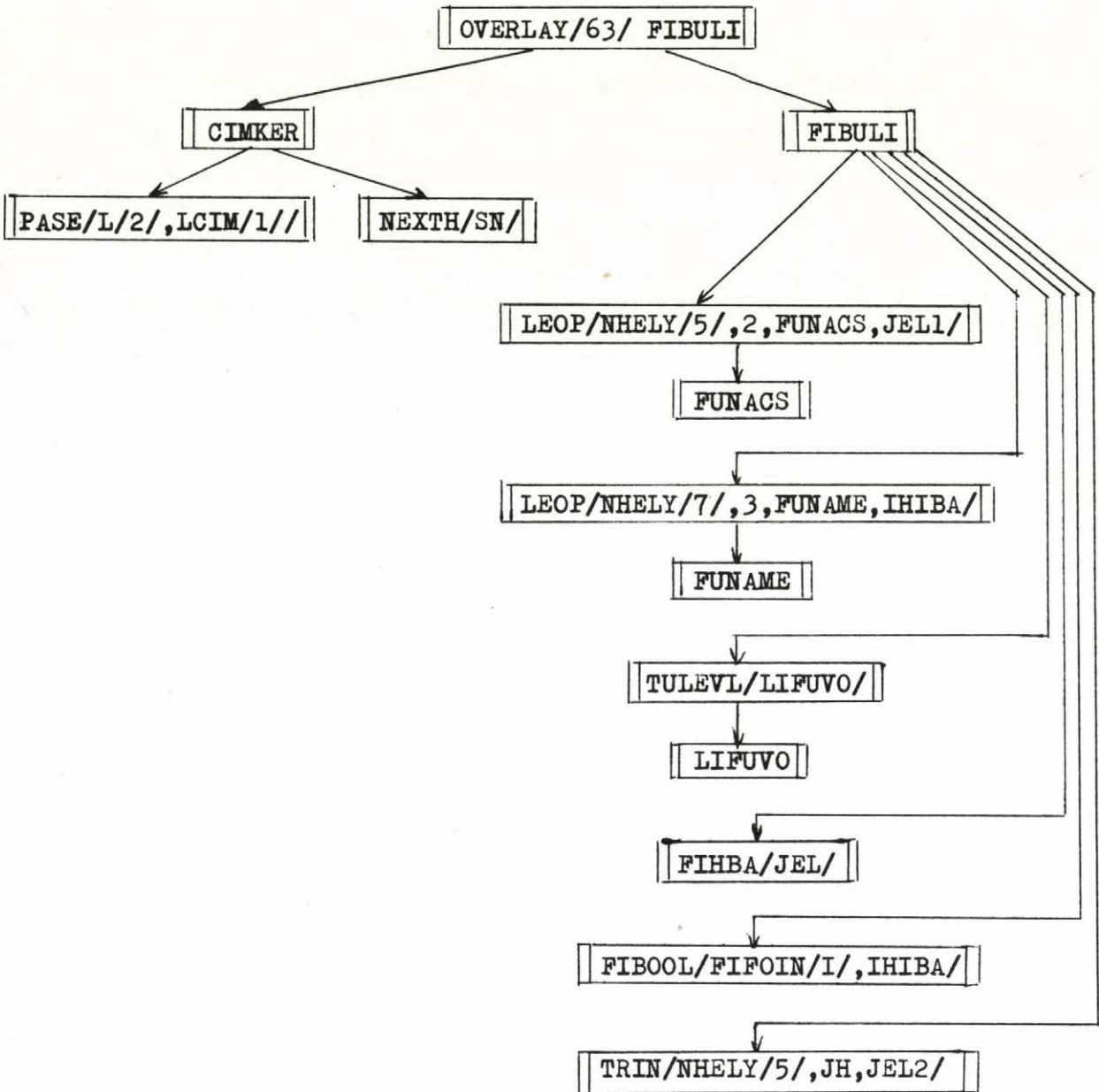
4. Ha a negáció alá bekapcsolt szögpont neve 0, akkor a hibaüzenet:

~~*****~~A NEGAACIO ARGUMENTUMA NEM FUEGGVEENYNEEV~~*****~~

Ha valamelyik Φ -Boole-függvény formulagráfja hibás, akkor egyetlen Φ -Boole-függvényre vonatkozó információ

em kerül elhelyezésre a FVCSOP szögpont alá.

- f. A TRIN szubrutin segítségével elhelyezésre kerülnek a FVCSOP alá a paraméterátadó közös területben leírt szögpontok.



21. ábra

3.2. A BOOLOP rutin által aktivizált, függvények feletti műveletek

3.2.1. HERT művelet

A Boole-függvények helyiértékes belső reprezentációját előállító program.

A program vezérlése az OVERLAY/16/ /22. ábra/ főprogramjával történik, amely a feladatot végrehajtó LINORG szubrutin aktivizálásán kívül az elhelyezett belső reprezentációra vonatkozó adatokat is kiírja.

A kiírás eredménye két táblázat. Az egyik táblázat elemei a létrehozott belső reprezentációkat tartalmazó szögpontok címei, a másiké azon formulák nevei, amelyek belső reprezentációja elkészült.

SUBROUTINE LINORG

A rutin a paraméterátadó közös terület /ARG/ nevű mezőjének RES tömbjébe elhelyezett információ alapján vezérli a feladat végrehajtását. A rutin alkalmas arra, hogy egyetlen aktivizálása során maximálisan 49 függvény belső reprezentációját készítse el. Ennyi függvénynév elhelyezését biz-

tosítja, ugyanis a RES tömb mérete.

A szubrutin belső reprezentáció elkészítéséhez a következő részfeladatokat végzi el.

- a. A MENTOE nevű változóba kimentti annak a szögpontnak a címét, ahová a belső reprezentációt el kell helyezni.
/LINTER alatti AND vagy OR/
- b. A RES tömbben elhelyezett formulák gyökérszögpontjainak neve vagy negatív címei alapján kikeresi a változólista címeket a VALTLIST illetve FORMDEF szögpont alól, és a FORMULA nevét vagy negatív gyökércímeket, a változólista gyökerének negatív címével helyettesíti, ha azt VALTLIST alatt találta, és a pozitívval, ha a FORMDEF alatt. Ha a VALTLIST és a FORMDEF szögpontok mindegyikét terminálisnak találta, akkor kiírja a

~~*****~~NINCS VAALTOZOLISTA.~~*****~~

üzenetet és megszakítja a program futását.

Ha a megadott függvénynek nincs változólistája

~~*****~~AZ n BELSOE NEVUE FUEGGVEENYNEK NINCS VAALTOZO
LITAAJA.~~*****~~

üzenetet írja ki és megszakítja a program futását.

A feladatot a SEGMENT/1/ részprogram hajtja végre.

- c. A RES-ben található változólista címek alapján kikeresi

a változó szögpontokat és azokhoz egy-egy kettő hatványt rendel, amelyet elhelyez a szögpont első attribútumában. A hozzárendelésnél figyelembe veszi az előforduló változók gyakoriságát. Ha a változók gyakoriságát rendre n_1, n_2, \dots, n_k -val jelöljük, amelyekre igaz, hogy $n_1 \geq n_2 \geq n_3 \dots \geq n_k$, akkor

$$\text{FREKVA}/n_i/ = 2^{i-1}$$

A feladatot több függvény esetén a SEGMENT/2/ részprogram, egy függvény esetén a VATOLT szubrutin hajtja végre.

d. A helyiértékes ábrázolású formula adminisztrálása a FORMDEF alatt.

- Ha a függvénynek a FORMDEF alatt még nincs változólistája, akkor azt a SEGMENT/7/ létrehozza és gyökérszögpontjának első attribútumába beír AND-et vagy OR-t aszerint, hogy a RES-ben mi volt az utolsó paraméter.
- Ha a függvénynek a FORMDEF alatt szerepel változólistája, akkor megvizsgálja, készült-e már belső reprezentáció az aktuális függvényről. Ha nem, akkor az első attribútumot feltölti. Ha igen, akkor a következő függvényre végrehajtja a d. feladatot.

e. Megkeresi az aktuális függvény formulagráfjának gyökércímét. Ha a függvényt nem találja, akkor kiírja az

n BELSO NEVU FUGGVENY NINCS A LISTABAN

üzenetet, és a műveletet megszakítja.

- f. Elkészíti a MENTOE változóban megadott szögpont alatt a formula belső reprezentációját. /FORMLO rutin/ Az elhelyezett formula negatív címével felülírja a RES-ben a függvény változólistájának gyökércímét és áttér, ha van ilyen, a RES-ben megadott további függvény feldolgozására, amely ismét a d. pontnál leírt művelettel kezdődik.

SUBROUTINE FORMLO/FOCIM,NCIM/

Egy adott függvény belső reprezentációját állítja elő.

A szubrutin által végrehajtott részfeladatok a következők:

- a. Megállapítja a FORMULA típusát a NOFOX szubrutin és a NOFOX által aktivizált SEGMENT/3/ részprogram segítségével. A formula típusát jellemző adatok a paraméterátadó közös terület /HIBA/ mezőjének IHIBA és NMUV nevű változójába kerülnek elhelyezésre.

Ha IHIBA > 0 A formula nem normálforma vagy olyan egytagu DNF illetve KNF, amelyben egy változónév többször is előfordul.

IHIBA=0	egyébként.
Ha NMUV=AND	egytagu DNF,
=OR	egytagu KNF,
--AND	KNF,
--OR	DNF,
=0	egytagu DNF illetve KNF, amelyben egy változó többször is előfordul.

Megjegyzés: Az egytagu DNF felfogható olyan KNF-nek, amelyben minden "diszjunkció" egytagu;
az egytagu KNF olyan DNF-nek, amelyben minden "konjunkció" egytényezős.

b. Elkészíti a belső reprezentációt vagy reprezentációkat a formula típusából és az NCIM bemenő paraméter értékétől függően. Az NCIM a rutin aktivizálásakor veszi át a MENTOE-ben elhelyezett értéket.

Ha a formula diszjunktív normál forma /DNF/ és NCIM az OR címe, elhelyezésre kerül a DNF belső reprezentációja. /LTREP szubrutin/

Ha a formula konjunktív normál forma /KNF/ és az NCIM az AND címe, elhelyezésre kerül a KNF belső reprezentációja. /LTREP szubrutin/

Ha a formula egytagu diszjunkció vagy konjunkció, NCIM értékétől függetlenül elhelyezi a LTREP a belső repre-

zentációt az NCIM-mel megadott szögpontra alá. Ha NCIM az AND címe, az egytagú diszjunkciót olyan KNF-nek értelmezi, amelyben minden "diszjunkció" egytagú.

Ha NCIM az OR címe, az egytagú konjunkciót olyan DNF-nek értelmezi, amelyben minden "konjunkció" egytényezős.

Ha a formula és NCIM az AND szögpontra címe, akkor a rutin a LTREP aktivizálásával elhelyezi a DNF belső reprezentációját az OR című szögpontra alá, és a LILIDU aktivizálásával a DNF-fel azonos KNF-át az AND szögpontra alá.

Ha a formula KNF és NCIM az OR szögpontra címe, akkor a rutin a LTREP aktivizálásával a belső reprezentációját a KNF-nek elhelyezi az AND szögpontra alá, és a LILIDU aktivizálásával a KNF-fel azonos DNF-át az OR szögpontra alá.

Ha a formula nem normál forma, akkor a formulával azonos DNF belső reprezentációja mindig elhelyezésre kerül az OR szögpontra alá a SEGMENT/8/ és a SEGMENT/10/ egymásutáni aktivizálásával.

Ha az NCIM értéke AND, a LILIDU szubrutin aktivizálásával a DNF-fel azonos KNF belső reprezentációja is elhelyezésre kerül.

SUBROUTINE NOFOX

A szubrutin a SEGMENT/3/-ban /12. ábra/ aktivizált bejárás /TOOPJ bejáró rutin/ után kialakított IAMUV és NMUV értékek alapján szövegesen kiírja a formula típusát.

A bejárás sikeréről a /MUNK/ közös adatmező JEL nevű változója tájékoztatja a rutint. Ha a JEL értéke 0-tól különböző, a bejárás nem sikerült, kiírásra kerül a

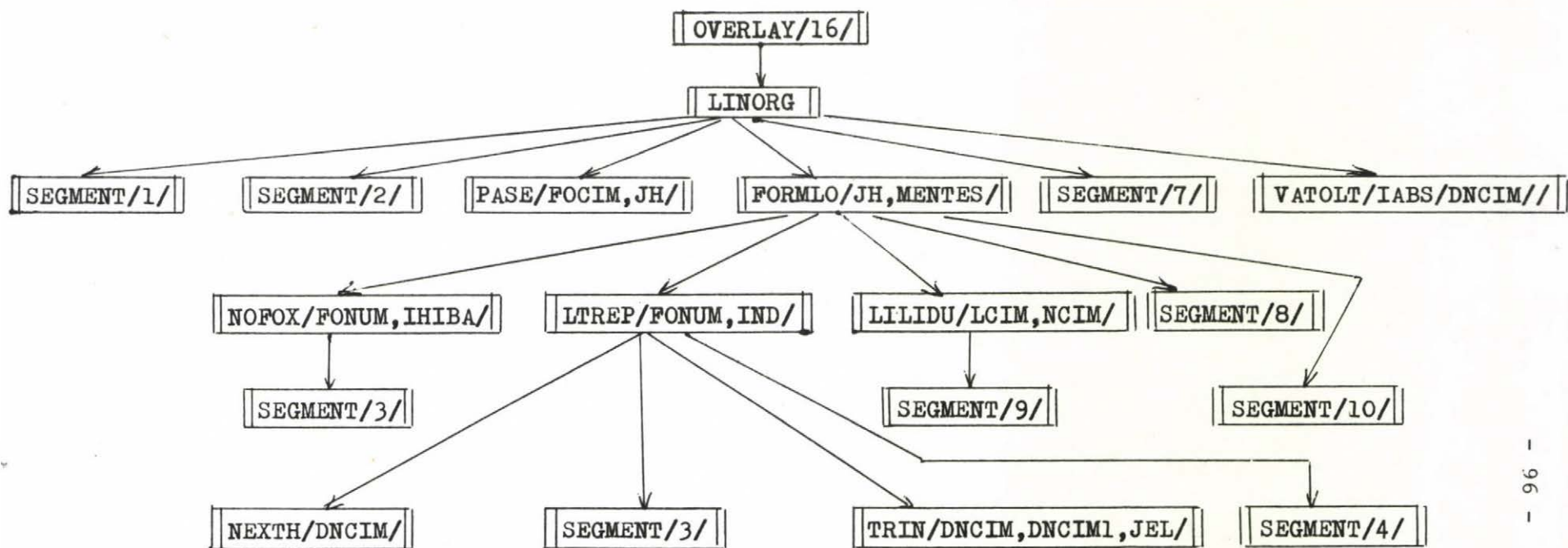
BEJARAS NEM SIKERULT

üzenet és a program futása megszakad.

A formulgráf bejárása során valamely konjunkcióban vagy diszjunkcióban szereplő különböző változókat kigyűjti egy munkamezőbe, és ha 23-nál több változót talál, kiírja a

TUL SOK VALTOZO

üzenetet és a program futását megszakítja.



22. ábra

SUBROUTINE FEJBEN

A FEJBEN rutin feladata /23. ábra/, hogy a formulagráf bejárása során kialakítsa a FORMLO rutin a. pontjában említett IHIBA és NMUV változók értékeit a formula típusának meghatározásához.

A formula típusát a formula változóinak /terminális elemek vagy az összetett név alapján terminálisnak tekintett elemek/ és a műveleti jeleknek a gyökérhez viszonyított szintszáma határozza meg.

SUBROUTINE LTREP/FVNEV,DNCIM/

A LTREP szubrutin /22. ábra/ a belső reprezentációt a függvény diszjunktív vagy konjunktív normálformájának formulagráfja alapján állítja elő a következőképpen:

- a. Megvizsgálja, hogy a két belső reprezentáció szerepel-e az AND vagy OR szögpont alatt. Ha szerepel, akkor a
n HELYERTEKES REPRESENTACIOJA MAR KORABBAN ELKESZULT
üzenettel a rutin munkája befejeződik.
- b. A rutin a SEGMENT/3/ segítségével /23. ábra/ aktivizálja a formulagráfot bejáró TOOPJ rutint, amely a FOBAN

external rutin minden szögponban való alkalmazásával biztosítja a belső reprezentációt leíró fa zárójeles formuláját.

c. A TRIN szubrutin bekapcsolja a belső reprezentációt a listába.

d. Ha a belső reprezentáció elhelyezését nem tudta végrehajtani, kiírja a

LINTER ALATT TRIN SIKERTELEN

üzenetet és leállítja a program futását.

e. A belső reprezentáció második szintjén lévő I / $1 \leq I \leq 23$ / nevű szögpontok közül törli azokat, amelyek egy korábban elhelyezett belső reprezentációban már szerepelnek. A törlés úgy történik, hogy a szögpontot a korábban elhelyezett I nevű szögpontra mutató SKIP elemmé változtatja. A művelet végrehajthatóságát az biztosítja, hogy az I nevű szögpont első előfordulásakor a /CIM/ közös adatmező INDEX nevű tömbjének INDEX/ I / elemébe elhelyezzük a szögpont címét. Az első előfordulást így az INDEX/ I /=0 feltétele biztosítja, és az INDEX/ I /≠0 érték adja meg a törléshez szükséges címet.

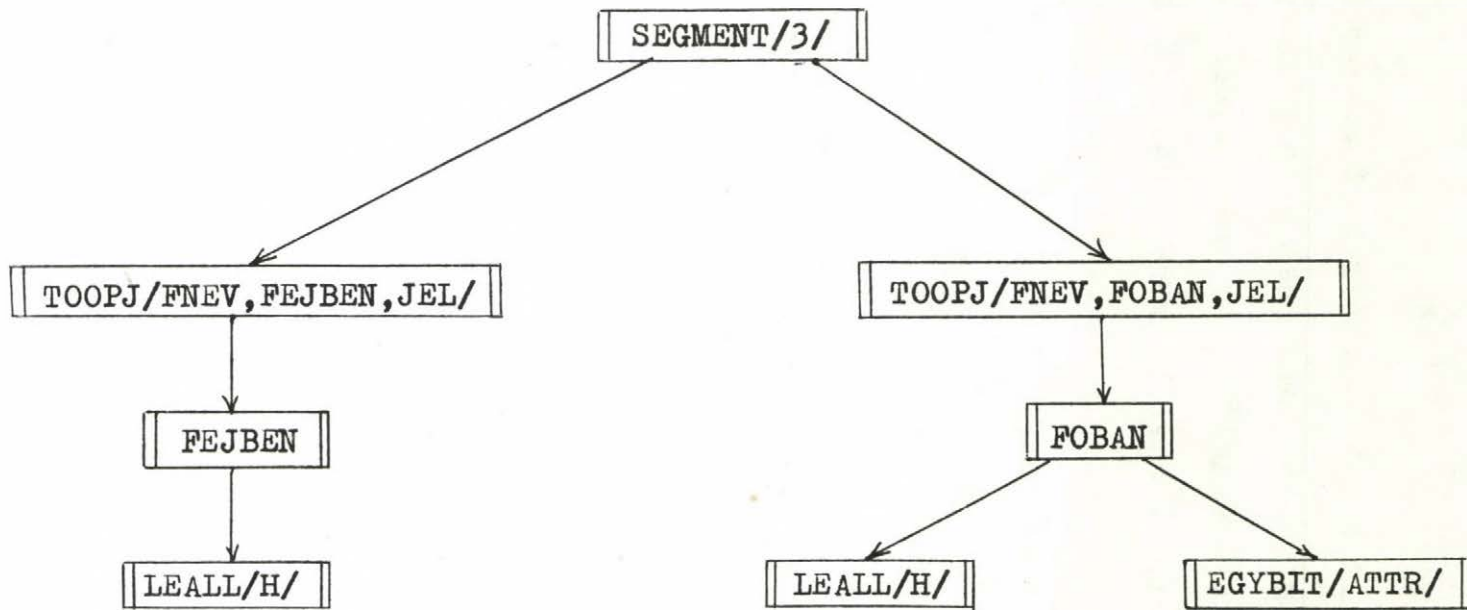
A feladat végrehajtását a SEGMENT/4/ vezérlőprogrammal aktivizált LEOPJ bejáró rutin végzi, amely minden szög-

pontban a REDUCE external rutin segítségével hajtja végre a fent leírt műveletet.

- f. Az először előforduló I nevű szögpontot összekapcsolja azzal a változó szögponttal, amelyhez az első attribútumban a 2^{I-1} értéket rendeltük. Az összekapcsoláshoz szükséges végpont címét, a /CIM/ közös adatmező CIM nevű tömbjében a CIM/I/ elemben a b. feladat végrehajtásakor helyezi el a rutin.

A szubrutin a FVNEV bemenő paraméterén keresztül kapja meg a feldolgozandó formula címét, és a DNCIM-en keresztül a LINTER alatti OR vagy AND címét. Ezeknek a paramétereknek FORMLO-ban a FONCIM illetve IND aktuális paraméterek felelnek meg.





23. ábra

SUBROUTINE LILIDU/LCIM,NCIM/

A rutin /22. ábra/ a DNF belső reprezentációjából a KNF belső reprezentációját /vagy fordítva/ állítja elő. A feladatot a SEGMENT/9/ vezérlőprogramja által aktivizált KNFKF szubrutinnal oldja meg.

SUBROUTINE KNFKF/IFV/

A rutin helyes működésének feltétele, hogy az IFV paraméter az átalakítandó belső reprezentációt tartalmazó fa gyökerének címe legyen.

A szubrutin a gyökérben elhelyezett konjunkciók illetve diszjunkciók felhasználásával, Voishvillo módszerét alkalmazva a diszjunktív szabályok alapján lehetséges műveletek elvégzésére elkészíti a formula diszjunktív illetve konjunktiiv normálformájának reprezentációját, és azt elhelyezi a /CIMES/ közös adatmező ICIM/18/ változójával meghatározott című szögpontra alá. ICIM/18/ a LINTER alatti OR illetve AND címét tartalmazza aszerint, hogy DNF illetve KNF az elhelyezendő belső reprezentáció.

Megjegyzés: az ICIM/18/-ba a műveletigénynek megfelelően a LILIDU helyezi el a kívánt címet.

SUBROUTINE SZAMIT/FVNEV/

A szubrutin strukturáját a 24. ábra mutatja.

A szubrutin alkalmas bármely olyan Boole-függvényt megadó formulával ekvivalens DNF elkészítésére, amely a negáció, diszjunkció, konjunkció, kizáró vagy, NOR, NAND műveletek jelén kívül más műveleti jelet nem tartalmaz. A feladat végrehajtása során a diszjunkciók beírása a /420/ nevű közös adatmező MATR/400,4/ nevű tömbjében/egy diszjunkció-egy sor/ alakulnak ki a kisebb indexek felé tömörített formában. A leírásban a formulagráf minden szögpontjának két bit szélességű mező felel meg. Ezekbe a mezőkbe kerülnek be a szögpontok logikai értékei. A 00 a határozatlan, a 01 a hamis, 11 az igaz logikai értéket ábrázolja. Az elmondottakból következik, hogy csak olyan formulagráf dolgozható fel, amely 48-nál több szögpontot nem tartalmaz. A szögponthoz tartozó műveleti jel és a szögponthoz rendelt mezőben található logikai érték meghatározza a művelet argumentumaihoz rendelhető logikai értékek lehetséges kombinációit. Ezek az érték kombinációk beírásra kerülnek a MATR tömb argumentumokhoz rendelt mezőibe, és egyidejűleg a művelet határozott értéke törlődik.

Ez az eljárás szemléletesen azt fejezi ki, hogy a formulagráf

valamely szögpontjában lévő logikai érték "átöröklődik" a szögpont alatt elhelyezkedő szögpontokra. Az átöröklődés minden lehetséges esete tárolódik a MATR tömbben. Az eljárás akkor ér véget, ha az "átöröklődés" folytán minden nem terminális szögpont határozatlan értékűvé válik, azaz a MATR tömbben csak a terminális vagy az összetett név alapján annak tekintett szögpontokhoz rendelt oszlopok tartalmazznak határozott logikai értéket.

A program végrehajtásának lépései a következők.

- a. A feldolgozandó formula neve alapján a rutin kikeresi a formulagráf gyökerének címét. Ha a keresés sikertelen, akkor kiírja a következő üzenetet:

```
*****AZ n NEVUE FORMULA NINCS A FORMULA SZOEGPONT  
ALATT*****
```

és a program futását megszakítja.

- b. A rutin bejárja a formulagráfot. /TOOP szubrutin/

Minden egyes szögpontban a TOLT rutin elhelyezi a szögpont címet a /420/ nevű közös adatmező ADR tömbjének soronkövetkező ADR/I/ elemébe. Az I index egyben a szögponthoz rendelt mező sorszámát a MATR tömbben. Ha a szögpontok száma 48-nál több, akkor a TOLT rutin kiírja a

~~*****~~TULCSORDULAAS ADR TOEMBBEN.~~*****~~

üzenetet és a program futását megszakítja.

- c. A MATR tömb első sorának első két bites mezőjébe beírja az igaz logikai értéket és a gyökér címéhez az ADR tömbben hozzáadja a vizsgálati jelet.
- d. Ha az aktuális szögpont nem terminális, vagy az összetett név komponensei alapján nem tekinthető annak, akkor végrehajtásra kerül a MATR tömb minden egyes sorában az átörökítés művelete. A műveletet megelőzi a műveletvégzéshez szükséges információk kigyűjtése a formulagráf-ból.

A szögpontban elhelyezett művelet argumentum leíró szögpontjainak címeit kigyűjtjük a /420/ nevű közös adatmező BE/10/ tömbjébe, amelynek első eleme az argumentumok számát BE/1/=K és BE/2/,...,BE/K+L/ a címeket tartalmazza. Ha $K > 9$, akkor kiírásra kerül a

~~*****~~TULCSORDULAAS BE TOEMBBEN.~~*****~~

üzenet és a program futása megszakad.

Az argumentumok címét helyettesítjük az ARG tömb azon elemének indexével, amelyben a szóbanforgó cím található.

Az átörökítés műveletét az aktuális szögpontban található

műveleti jelnek megfelelő szubrutinokkal végezzük. Az ADR tömbbe a művelethez tartozó oszlopokba elhelyezzük a vizsgálati jeleket.

Az aktuális szögpont Z oszlopát minden sorban megvizsgáljuk és onnan töröljük a határozott értékeket. Ennek folytán egyező sorokat kaphatunk, amelyek közül elegendő egyet megtartani. A fölösleges sorokat a NYELOE rutinnal töröljük. /a+az=a/ Ekkor a Z oszlopbeli művelettel az aktuális sorban nem foglalkozunk, hanem rátérünk a következő sorra. A mennyiben a Z oszlopbeli műveletet fel kell dolgozni, akkor a BE tömbben tárolt oszlopszámok felhasználásával vagy beírjuk az aktuális sorba az argumentumok egyetlen lehetséges kombinációját vagy az aktuális sort annyiszor másoljuk le MATR-ba, amennyi a lehetséges kombinációk száma, és mindegyik sorban az argumentumok oszlopait feltöltjük egy-egy lehetséges kombinációval.

Ha a sorok átmásolása során a MATR tömb betelt, és újabb sor átmásolása lehetetlen, kiírásra kerül a

TULCSORDULAAS MATR-BAN

MATR SORSZAAMA: n OSZLOPSZAAMA: m

üzenet és IMAX változó felveszi a -400 értéket.

A SZAMIT rutin, ha IMAX értékét negatívnak találja, visszaadja a vezérlést FORMLO közvetítésével LINORG-nak, amely ebben az esetben, ha van még feldolgozandó függvény áttér annak feldolgozására, egyébként visszaadja a vezérlést BOOLOP-nak.

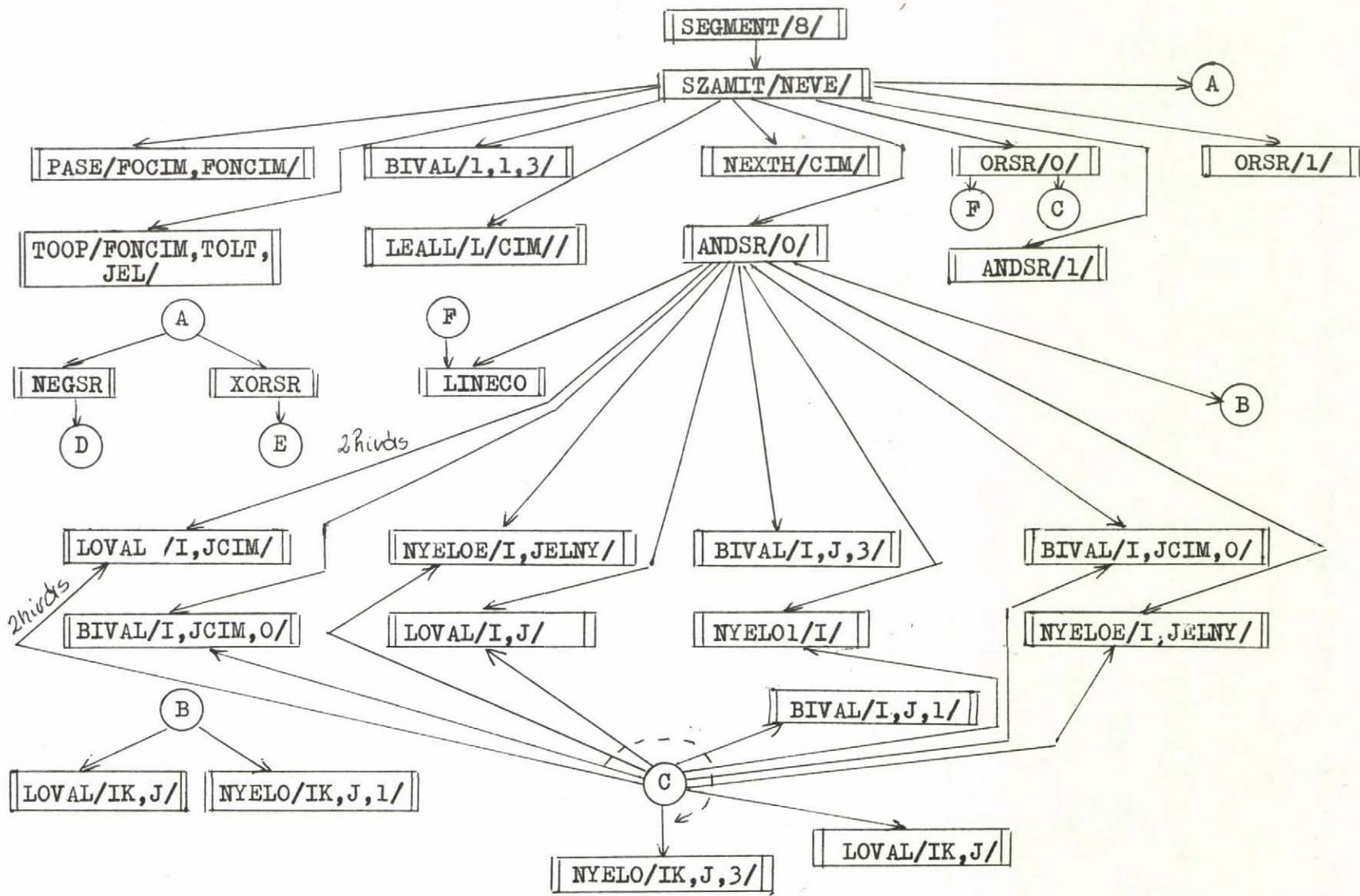
E feltöltés után összevonjuk az egyező vagy az egy értékben különböző sorokat $/a+a=a, a+ax=a, ax+a\bar{x}=a/$ a NYELO illetve NYELO1 rutinokkal aszerint, hogy a beírt érték-kombináció egyetlen illetve több határozott logikai értéket tartalmaz.

Olyan érték-kombináció beírása nem lehetséges, amelyben van valamely argumentum korábbi logikai értékével ellenkező érték. Az ilyen sort a MATR-ból töröljük. Ha a törlés után a MATR tömb kiürült, akkor a formula azonosan hamis. Ilyenkor belső reprezentáció helyett a formula nevével egyező nevű szögpontra kerül elhelyezésre, amely attribútumában a hamis logikai értéket tartalmazza.

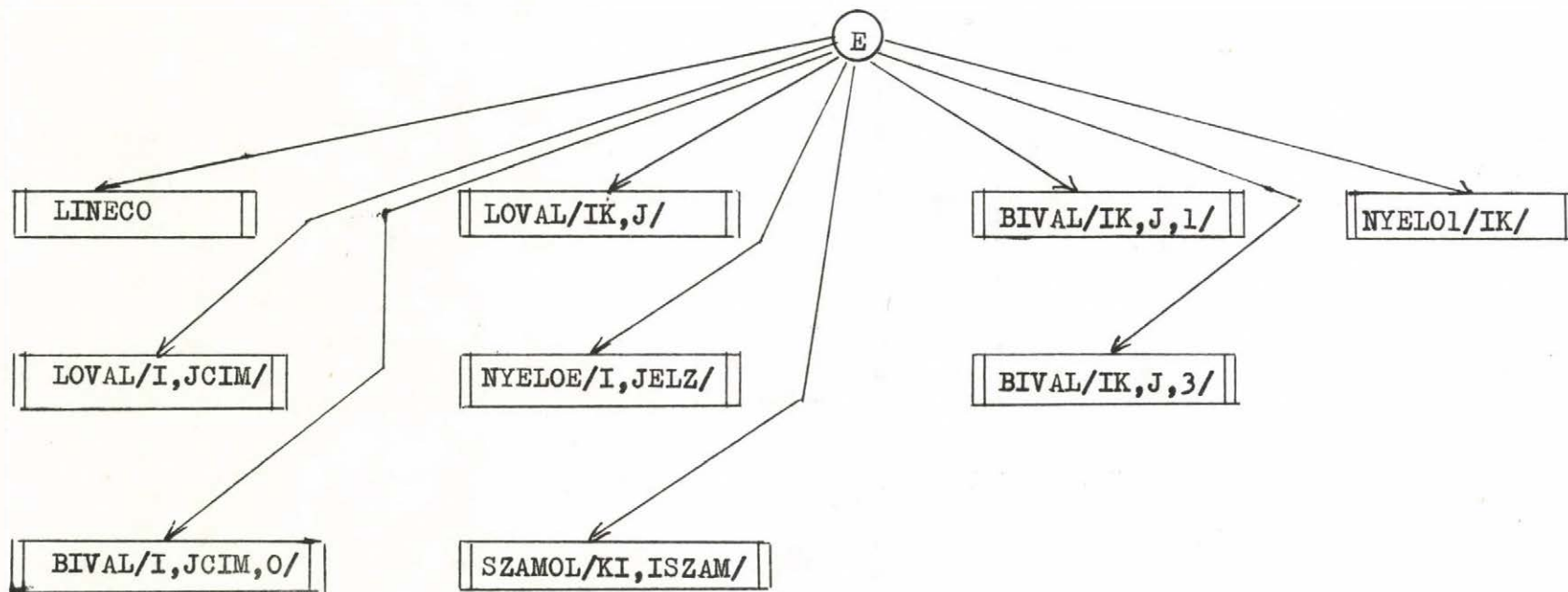
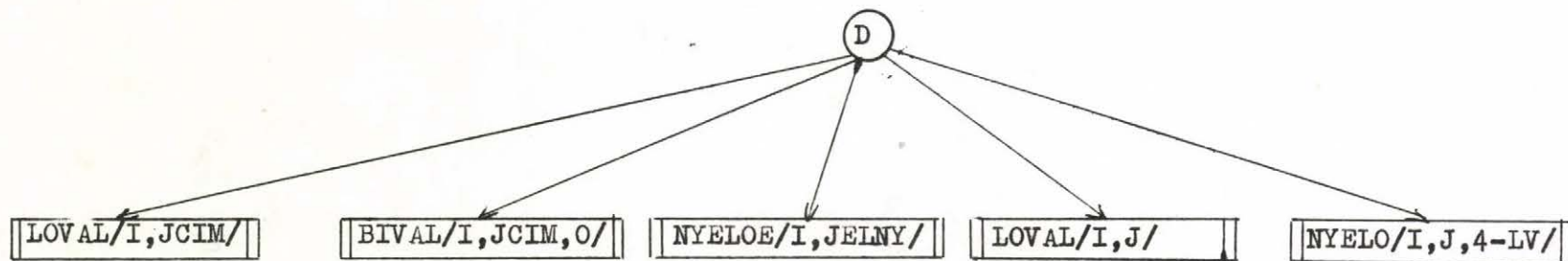
Ha az összevonások és a határozott logikai értékek törlése során minden oszlopba határozatlan érték kerül, akkor a formula a változók minden érték-kombinációjára igaz /azonosan igaz/, így az előbb említett szögpontra kerül elhelyezésre, de attribútumában igaz értékkel.

Ha az aktuális szögpontra terminális, akkor az ADR tömb

szögponthoz tartozó oszlopába elhelyezett vizsgálati jelet töröljük. Ha már nincs vizsgálati jelet tartalmazó elem az ADR tömbben, akkor a MATR tömb feltöltése befejeződött, a rutin befejezi működését, egyébként a feladat a vizsgálati jelet tartalmazó elemhez tartozó szögponttal folytatódik, azaz újra kezdődik a d. pontban leírt folyamat.



24. ábra



24. ábra folytatás

Az átörökítést végző rutinok

SUBROUTINE ANDSR/INDEX/

A rutin a konjunkció vagy NAND műveleti jel esetén ír az argumentumok helyébe értékeket.

Ha a művelet konjunkció és a művelet értéke igaz vagy a művelet NAND és a művelet értéke hamis, akkor minden argumentum igaz lesz. Egyébként annyi argumentumkombinációt generál, amennyi az argumentumok száma. Egy argumentumkombinációba egy argumentum értéke hamis, a többié határozatlan.

SUBROUTINE ORSR/INDEX/

A rutin a diszjunkció és a NOR műveleti jel esetén ír az argumentumok helyébe logikai értékeket.

Ha a művelet diszjunkció és a művelet értéke hamis vagy a művelet NOR és a művelet értéke igaz, akkor minden argumentum hamis lesz. A másik két esetben annyi argumentumkombinációt generál, amennyi az argumentumok száma. Itt egy argumentum értéke igaz, a többié határozatlan.

SUBROUTINE NEGSR

A rutin a negáció műveleti jel esetén generálja a műveleti jel logikai értékével ellentétes értéket az argumentumba.

SUBROUTINE XORSR

A rutin a kizáró vagy műveleti jel esetén ír az argumentumok helyére logikai értékeket.

Ha az argumentumok száma n és a művelet logikai értéke igaz illetve hamis, az összes páros számú illetve páratlan számú igaz értéket tartalmazó értékkombinációt generálja a rutin.

$$\left(\sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} \binom{n}{2i} \text{ illetve } \sum_{i=0}^{\lfloor \frac{n}{2} - 0.5 \rfloor} \binom{n}{2i+1} \text{ db.} \right)$$

A sorok összevonását végző rutinok

SUBROUTINE NYELOE/I, JELZOE/

A rutin összehasonlítja az aktuális sort a MATR tömb többi 0-tól különböző sorával és ha egyezőt talál, akkor azt törli.

A JELZOE értékét 0-tól különbözőre változtatja. Ennek alapján dönt a műveletvégző rutin az átörökítés elvégzéséről.

SUBROUTINE NYELO/I,J,LV/

A rutin akkor aktivizálódik, amikor az átmásolt sor egyetlen mezőjébe kell határozott logikai értéket beírni.

A rutin által végrehajtott részfeladatok a következők.

Megvizsgálja, hogy van-e az átmásolt sorral azonos sor, ha ilyet talál, akkor azt törli. /a+ax=a/

Megvizsgálja, hogy van-e egyező sor, ha az átmásolt sor egyetlen mezőjébe a logikai értéket írja be. Ha talál ilyen sort, akkor a másik /a+a=a illetve $ax+a\bar{x}=a$ / vele egyező sort törli.

SUBROUTINE NYELO1/I/

A rutin akkor aktivizálódik, amikor az aktuális sorba, az aktuális művelet minden egyes argumentumához tartozó mezőjébe határozott logikai értékek kerülnek beírásra.

A rutin megvizsgálja, van-e az aktuális sorral egyező sor és ha van, ezt a sort törli.

A rutin rendre kicseréli az aktuális sor szóbanforgó mezőiben a logikai értékeket az ellentetjükre, és minden csere után megvizsgálja, van-e az aktuális sorral egyező sor, ha van, akkor az aktuális sorból a vizsgált logikai értéket törli.

Kiolvasás és a beírás a MATR tömbbe

A beírás műveletét a BIVAL/I,J,LV/ szubrutin végzi. Az I,J bemenő paraméterek adják a mező helyét a MATR tömbben, I a MATR tömb sorindexe, J a mező sorszám. LV bemenő paraméter az a logikai érték, amelyet be akarunk iratni.

A kiolvasás műveletét a LOVAL/I,J,LV/ szubrutin végzi, ahol I,J bemenő paraméterek jelentése azonos az előbbivel. LV kimenő paraméter a keresett logikai értéket adja meg. A kiolvasás után az adott mezőben a logikai érték megmarad.

3.2.2. CHELY művelet

Logikai értékek behelyettesítésével nyert új Boole-függvény formulagráfját és változólistáját, illetve a függvényértéket előállító program.

A feladatot az OVERLAY/17/ vezérlőprogramjával aktivizált KOHEFO rutin oldja meg.

SUBROUTINE KOHEFO

A rutin /25. ábra/ működése a következő részekre tagolható.

- a. /RES/ közös adatmezőben elhelyezett adatokból meghatározza a formula nevét és a formulába helyettesítendő változók számát.
- b. Kikeresi a listából a formula gyökércímét /PASE szubrutin/, és azt megőrzi a JBLOP nevű változóban.
- c. Bejárja a formulagráfot, és rendre a RES-ben megadott nevű változó szögpontok második attribútumaiba elhelyezi a b_i név mellett megadott c_i logikai értéket.
Ha a c_i érték igaz, akkor az attribútumb^ua 1, ha hamis a -1 érték kerül. /TOOP rutin és a KERES external rutin/

- d. Műveletvégzés a konstansok behelyettesítése után. A formulagráf endorder bejárása után /ENDORD rutin/, a műveletek argumentumainak értékei alapján meghatározza és az illető művelet attributumába elhelyezi a művelet eredményét, vagy azt az információt, amely az új formulagráf kialakításához szükséges. /VOPKO external rutin/
- e. Ha a függvényérték kiszámítása volt a cél $/a_1=a_2$ és minden változót konstanssal helyettesítünk/, akkor a gyökérszögponthoz kialakult eredményt a RES/IRES+1/-ben őrizzük meg, és a feladat befejeződik.
- f. Ha célunk új formula létrehozása volt $/a_1 \neq a_2/$, akkor a TOOPJ szubrutinnal bejárja a régi formulát és a szögpontokban a GRCOP rutin aktivizálásával létrehozza az új formulagráfot leíró zárójeles formulát.
- g. Elhelyezzük a formulagráfot a FORMULA szögpont alá. Ha az elhelyezés sikertelen volt, kiírjuk a
TRIN-BEN HIBA
üzenetet és a program futását megszakítjuk.
- h. Az új formula változólistáját létrehozzuk a FLIST szubrutin segítségével.
- i. Bejárjuk a régi formula gráfját, és a második attributumokban elhelyezett logikai értékeket töröljük.

SUBROUTINE VOPKO

A VOPKO rutin működése függ az érintett szögpont típusától.

- a. Ha a szögpont terminális, a /TEWE/ közös adatmező LUK nevű veremébe elhelyezzük a szögpont attribútumában tárolt logikai értéket.
- b. Ha a szögpont nem terminális, és az AND,OR,NEG,NOR,NAND műveleti jelek valamelyikét tartalmazza, akkor aktivizáljuk a megfelelő műveletkiértékelő rutint.
- c. Ha a műveleti jel nincs a b. pontban felsoroltak között, akkor kiírásra kerül a
*******A MUEVELET NEM ELVEEGEZHETOE.*******
A MUEVELET BELSOE NEVE: n
üzenet és a program futása befejeződik.
- d. Elhelyezi a művelet eredményét a szögpont második attribútumába és a LUK verembe.

Műveletkiértékelő rutinok

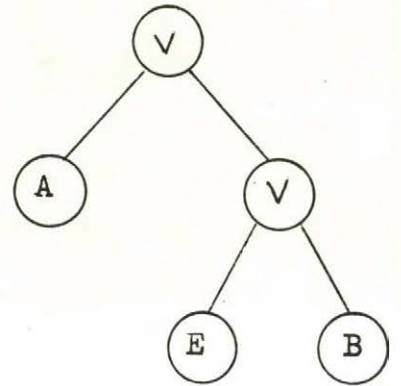
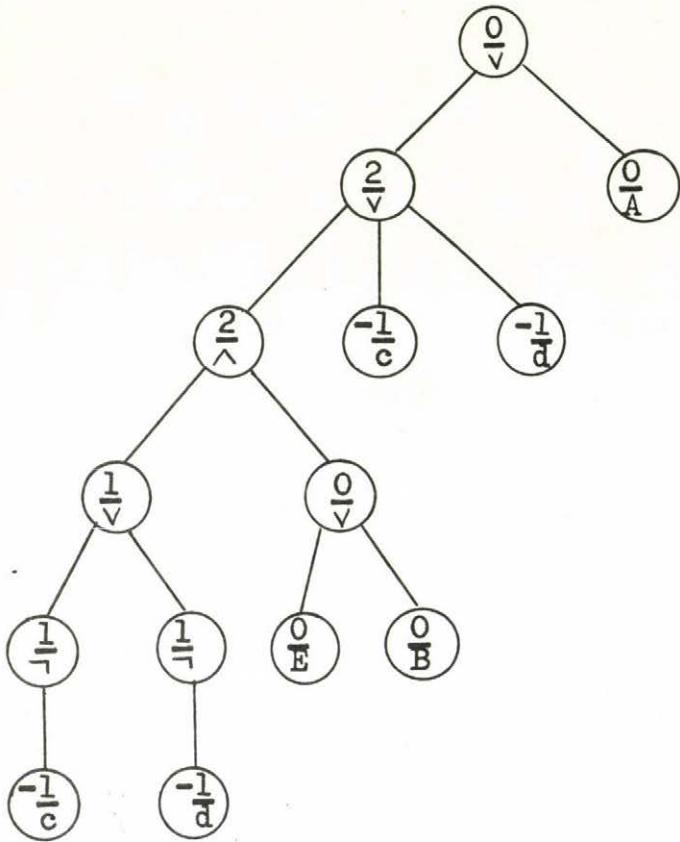
A rutinok a művelet argumentumainak értékei alapján kiszámítják a művelet értékét. A művelet értéke a következő lehet:

- 1 ha a művelet eredménye igaz,
- 1 ha a művelet eredménye hamis,
- 0 ha a művelet eredménye határozatlan,
- 2 ha a művelet eredménye határozatlan egyetlen nem meghatározott argumentum miatt.

Az argumentumszám és a művelet eredménye a rutinok formális paraméterei.

SUBROUTINE GRCOP

A szubrutin egy olyan formulagráf során végez műveleteket a szögpontokban, ahol a műveleti szögpontokban szerepel a művelet értéke is /1,-1,0,2/, és a cél a gráfnak megfelelő zárójeles formula előállítás. Ez azt jelenti, hogy a műveleti értékek alapján az eredeti gráfot egyszerűsíteni kell. Tehát a gyökértől eltekintve elhagyjuk azokat a részfákat, amelyeknek az igaz vagy a hamis a gyökérben a művelet értéke. Elhagyjuk a 2 műveleti értékű szögpontok igaz és hamis értékű argumentumainak megfelelő szögpontokat, és ugyanitt a műveleti jelet megfelelően megváltoztatjuk. Illusztrációként a 34. ábrán egy ilyen átalakítás látható.



f/a,b,c,d,e/=a v //c v d /∧/ e v b /v c v d //

f/a,b,0,0,e/=a v /e v b /

34. ábra

A szubrutin által a szögpontban végzett műveletek függenek a szögpont második attributumában szereplő műveleti értéktől, a bejárásban előtte lévő szögponthoz viszonyított helyzetétől és attól, hogy a gráf gyökere-e a vizsgált szögpont.

A rutin a szögpontokban kétféle műveletet hajt végre

- zárójelek elhelyezése
- egyéb szimbólumok elhelyezése.

A rutin először a szögpont és az előzőleg érintett szögpont helyzetét vizsgálja. /A kezdeti szint a 0./ Legyen a vizsgált szögpont szintszáma n , az előzőleg érintetté k .

- ha $n=k$ nem kell zárójel
- ha $n > k$ és a műveleti érték 0, akkor egy nyitózárosjelet kell kitenni.
- ha $n < k$ akkor $k-n+1-n_2$ záró zárójel kell kitenni, ahol n_2 a két szögpont közötti uton a 2-es műveleti értékű szögpontok száma.

A rutin másodszor a második attributumában lévő m /műveleti/ értéket vizsgálja. A két vizsgálat eredményeként alakul a szögpontban végzett művelet.

Legyen $m=1$ vagy -1 /konstans/

- ha $n=1$ /a gyökérben vagyunk/, akkor leírja az a_2 nevet tartalmazó konstans függvény formulagráfjának zárójeles formuláját, és nem végez több műveletet.
- ha $n \neq 1$, akkor nem tesz ki zárójelet, és ezen az ágon nem folytatja a bejárást.

Legyen $m=0$ /határozatlan/

- ha $n > k$, az előzőleg érintett szögpontban $m \neq 2$, de a művelet NAND illetve NOR, akkor kitesszük a nyitó zárójelet. Ezután a szögpont leírása következik - szögpont neve és attribútumai.
- ha $n < k$, akkor az $n-k+1-n_2$ záró zárójel és a szögpontleírás kerül be a zárójeles formulába.

Legyen $m=2$

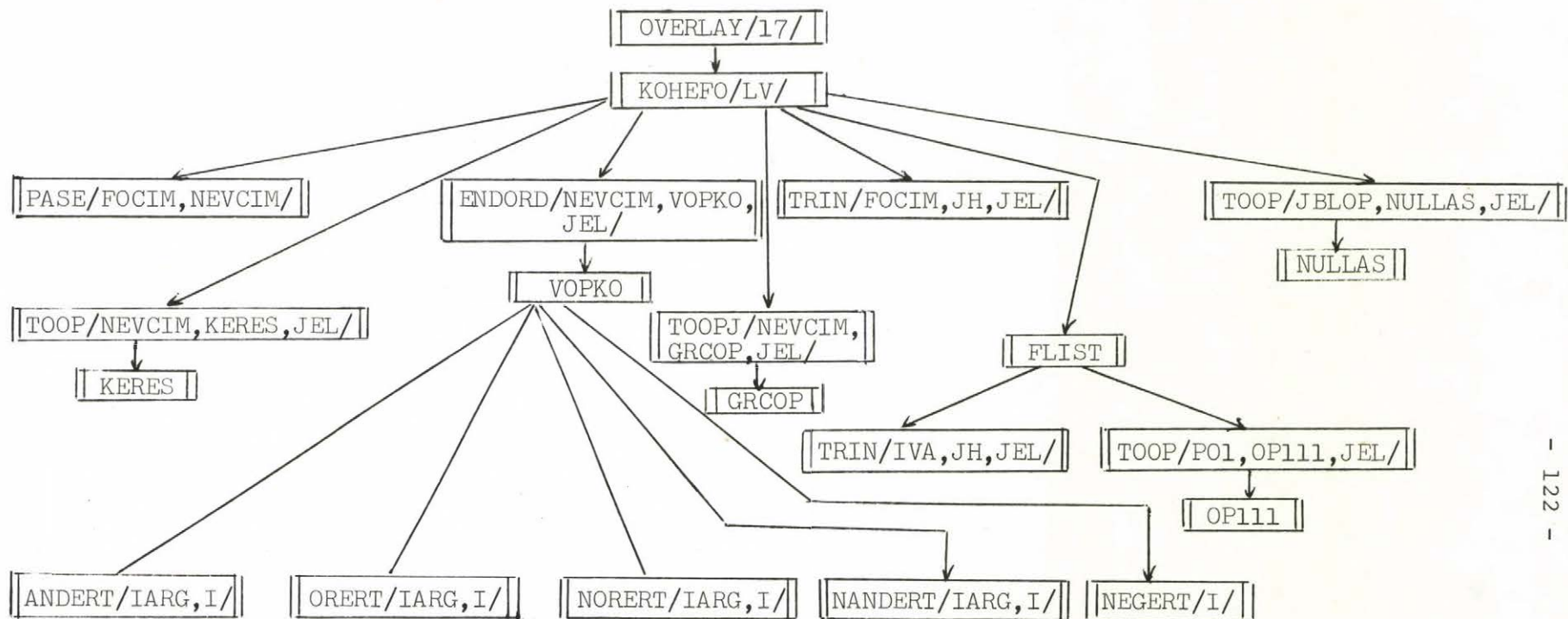
- ha $n > k$, akkor a zárójel elhelyezésre az $m=0$ esetben elmondottak érvényesek.

Amennyiben a művelet AND vagy OR, akkor a szögpontot nem írjuk le. A munkamezőbe beírjuk a szögpont szintszámát és a MUNK változóban jelezzük, hogy a leírás utolsó karaktere nyitózárlójel.

Ha a művelet NAND vagy NOR, akkor a szögpontot negáció műveleti jellel írjuk le.

Az $n=1$ esetben a szögpont neve a_2 lesz.

- ha $n < k$, akkor a zárójel elhelyezésre az $m=0$ esetben elmondottak érvényesek. A szögpont elhelyezés pedig a fent elmondottak szerint történik.



25. ábra

3.2.3. EGYSZ művelet

Boole-függvények egyszerűsítését megvalósító program.

A program vezérlését az OVERLAY/10/ főprogramja végzi. A feladatmegoldás magába foglalja a helyértékes reprezentáció előállítását, ha az még nem létezik. Az egyszerűsítés műveletét a [4]-ben leírt algoritmus alapján végzi el, és az eredményt kiírja a sornyomtatóra.

3.2.4. MIN művelet

Boole-függvények együttes minimalizálását megvalósító program.

A program vezérlését az OVERLAY/9/ főprogramja végzi. A programra az EGYSZ műveletnél elmondottak érvényesek.

3.2.5. BDIFF művelet

A program vezérlését az OVERLAY/12/ főprogramja végzi. A feladatmegoldáshoz a helyértékes reprezentációt előállítja, ha az még nem adott.

A BDIFF művelet az f függvényből előállítja $f_{x=1}$ és $f_{x=0}$ függvényeket és képezi a $XOR/f_{x=1}, f_{x=0}/$ függvényt.

3.2.6. XOR művelet

A program vezérlését itt is az OVERLAY/12/ főprogramja végzi oly módon, hogy adott f,g függvényekkel közvetlenül aktivizálja a XOR/f,g/ függvényt előállító eljárást.

4. A Boole-függvény kezelő rendszer aktivizálása

A továbbiak megértésének érdekében megadjuk azon file-ok listáját, amelyekre a leírás során hivatkozunk.

név	dsi	disc	funkció
OVERLAYLAPOK	FOE	8	Az OVERLAY lapok és szegmensek vezérlő programjai.
LIBBFKR8	28	8	A segédkönyvtár tartalmazza a LIDI-72 rutinjait és a rendszerhez tartozó szubrutinokat.
DIRBFK8	29	8	A könyvtár directory-je.
LISTAADAT	ADFI	8	Főprogram változat.
GYOEKEERADAT	GYOK	8	Főprogram változat.
GYOEKERNINCS	NOTI	8	Főprogram változat.
TASKNAME	1111	8	A gyökérlista.
	8000	8	A felhasználó file-ja, ahol a feladat végrehajtása után az eredménylista megőrződik.
BOOLEABSZTASZK1609	15	9	} Helyértékes belső reprezentációt előállító programok változatai.
BOOLEABSZTASZK1609	15	9	
BOOLEFKZELOER9	BFKR	9	Az összes OVERLAY lap és SEGMENT tárolása kártyakép alakjában.

név	dsi	disc	funkció
SYLOFUS	LOFU	8	Az első osztályba tartozó műveletvégző lapok tárolása bináris alakban /a főlap kivétel/
GYOKEERINPUT8	GYOK	8	} A főlap változatainak tárolása bináris alakban.
ADATFILEINPUT	ADFI	8	
LINTERORS	ORAN	8	A függvény helyiértékes belső reprezentációját előállító program tárolása bináris alakban.
HELYETTESITES8	HELY	8	Megadott logikai értékek behelyettesítésével nyert új függvényt előállító program tárolása bináris alakban.

Az IDIL27 működési módja az I bemenő paramétertől függően 3 féle lehet, ennek megfelelően a főprogramnak három változata van, az egyes változatok,

ha I=0 a dsi=ADFI,
I=1 a dsi=NOTI,
I=2 a dsi=GYOK

azonosítóju bináris file-okon helyezkednek el.

A többi lap vezérlő programjai a dsi=FOE azonosítóju file-on a lapok által aktivizált szubrutinok és a LIDI 72 rendszer rutinjai a dsi=28 és dsi=29 azonosítóju file-okon helyezkednek el. Az OVERLAY program konszolidálás után a dsi=15 azonosítóju SOCR file-ra kerül.

A könyvtár nyitó kártyái a következők:

```
DEF/O,W,29,21C111,DIRBFK8,1,,I/
```

```
DEF/O,W,28,21C111,LIBBFKR8,1,,I/
```

Az ADFI file nyitó kártyája:

```
DEF/O,W,ADFI,21C228,LISTAADAT,1,,I/
```

A NOTI file nyitó kártyája:

```
DEF/O,W,NOTI,21C228,GYOEKERNINCS,1,,O/
```

A GYOK file nyitó kártyája:

```
DEF/O,W,GYOK,21C228,GYOEKEERADAT,1,,I/
```

A FOE file nyitó kártyája:

```
$DEF/O,W,FOE,21C228,OVERLAYLAPOK,1,,O/
```

A SOCR file nyitó kártyája:

```
$SOCR/A,15,1280,500/
```

Az OVERLAY programok aktivizálása a leirtaknak megfelelően a következő RLDR kártyákkal történhet:

```
I=0 $RLDR/MIEZ,15,ADFI,FOE/
```

```
I=1 $RLDR/MIEZ,15,NOTI,FOE/
```

```
I=2 $RLDR/MIEZ,15,GYOK,FOE/
```

Ezt követi közvetlenül a fent említett AUX kártya és a

```
$MIEZ,15 task name kártya. A konszolidált program ugyanis
```

```
$SOCR/A,15,1280,500/ kártyával megnyitott file-on található.
```

Az elmondottaknak megfelelően az OVERLAY program három változatának végrehajtása a következő programokkal biztosítható:

I=0

\$RLDR/MIEZ,15,ADFI,FOE/
\$AUX,28,29
\$MIEZ,15

I=1

\$RLDR/MIEZ,15,NOTI,FOE/
\$AUX,28,29
\$MIEZ,15

I=2

\$RLDR/MIEZ,15,GYOK,FOE/
\$AUX,28,29
\$MIEZ,15

Egyéb felhasználói igények kielégítésére létrehoztunk más aktivizálási lehetőségeket is.

1. Gyakran előforduló feladat a LIDI 72 bemenő nyelvén leírt Boole-függvény rendszerek helyiértékes belső reprezentációjának előállítás. A feladatot végrehajtó programcsomag abszolút task formájában helyeztük el a `BOOLOABSZTASZK1609` illetve a `BOOLLABSZTASZK1609` azonosítóju file-on. Előbbi abszolút taskot akkor használjuk, ha a lista inicializálása a gyökér bevételel történik és utóbbit, ha korábban előállított listát akarunk a feladattal bővíteni vagy módosítani.

A működéshez szükséges program a következő:

```
$DEF/O,W,15,21C228,BOOL(Ø)ABSZTASZK1609,1,,I/
      {1}
```

A gyökért tartalmazó /dsi=1111/ file nyitó kártyája, ha BOOLØABSZTASZK1609-en tárolt feladatot akarja végrehajtani.

A listát tartalmazó 8000-es file nyitó kártyája /írásra is megnyitva!/:

```
$TNEV,15      task név kártya
```

Mivel a dsi=28,29 file-okon elhelyezkedő programkönyvtár a LIDI 72 rutinjain kívül a rendszer rutinjait is tartalmazza, létrehoztunk egy olyan változatot is, amely az eredeti csak LIDI rutinokat tartalmazó programkönyvtár használatát is lehetővé teszi.

2. Az egyes feladatmodulok bináris file-okon vannak elhelyezve.

Ha a felhasználó csak adatstruktúra kialakító műveleteket akar végrehajtani, akkor a bináris programok összeszerkesztését az

\$RLDR/TNEV,15,{ADFI,LOFU/
 }GYOK}

vezérlőkártyával végezheti. A szükséges nyitó kártyák

\$DEF/O,W,LOFU,21C228,SYLOFU8,1,,I/

{ \$DEF/O,W,ADFI,21C228,ADATFILEINPUT,1,,I/ }

{ \$DEF/O,W,GYOK,21C228,GYOKERINPUT8,1,,I/ }

{ } -ben a vagylagos lehetőséget jelöljük.

A feladat indítható tárolt és gyökér listával is.

Ha ^a felhasználó helyiértékes belső reprezentációt is elő akar állíttatni, akkor az RLDR kártyán a file lista negyedik eleme ORAN, és a szükséges nyitó kártya:

\$DEF/O,W,ORAN,21C228,LINTERORS,1,,I/

Ha a felhasználó változó helyettesítést akar végrehajtani, akkor az RLDR kártyán a file lista 4. eleme HELY, és a szükséges nyitó kártya:

\$DEF/O,W,HELY,21C228,HELYETTESITES8,1,,I/

3. A teljes feladatot COSY file-on tároltuk. Ha a felhasználó erről a file-ról akarja aktivizálni a feladatot, akkor a feladat különböző változatainak megfelelően a következő COSY deck-eket kell összeállítani.

a. Ha csak az adatstruktúra kialakító műveleteket akarjuk végrehajtani, akkor az összeállítás a következő:

SCOSY

BAGOLYO DECK/ I=BFKR,H,G=FTNU

MINDENKI

BOOLEMUV

KOEZOES

GYOKERBE

BAGOLY4

INDITO

BAGOLY1

FUGVAL

HIBJELEZ

POLISHKA

VARLISTA

BAGOLY2

POKHAALO

BAGOLY3

FIHEZ

ENDCOSY/

/Mindegyik kártyán a fel nem tüntetett szövegrész az első kártyával azonos./

- b. Ha a felhasználó kéri a belső reprezentáció előállítását is, akkor ki kell bővíteni a fenti sorozatot a következő kártyákkal:

LINORG

SEGEDO

FORMULA

LIFADU

FORMULA1

FORMULA2

FORMULA3

SEGED2

SEGED3

KIFEJT

SEGED1

PE9

- c. Ha a felhasználó kéri a logikai értékek helyettesítésével előálló Boole-függvény rendszer előállítását, akkor a bővítés a következő DECK kártyákkal történik.

KONSTANS

SEETA

VARLISTA

SEGED2

A működéshez szükséges további kártyák:

\$FTNU/I=SHO,X,N,D,S/

\$RLDR/MIEZ,15,LGO/

\$AUX, { 4,5 }
 { 28,29 }

\$MIEZ,15

Nyitó kártyák:

\$SOCR/A,15,1280,500/

\$DEF/O,W,BFKR,21C228,BOOLEFKEZELOER9,Ø1,,I/

5. Példák a BFKR működésére

1. Adatmegadás

a. $\boxed{/\text{HERT}/F11,F12,F13,F14,F15,\text{OR}/}$

$\boxed{\text{PATH BOOLEFV, LEIRAS, MUVELET}}$

b. $\boxed{\text{vvvvvvv } F15=A}$

$\boxed{F14=B_{\text{N}}-C_{\text{N}}D}$

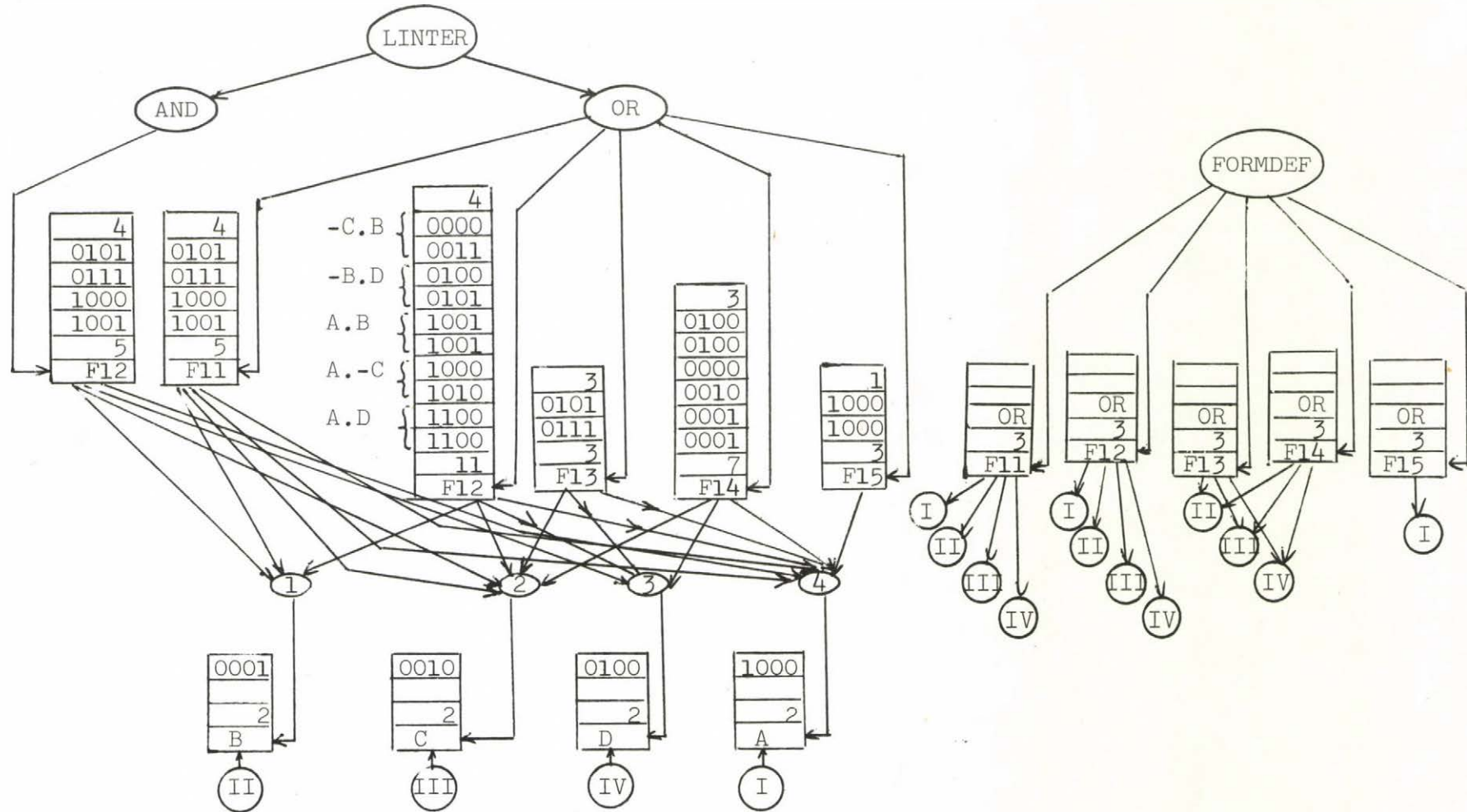
$\boxed{F12=/A+-B/_{\text{N}}/B+-C+D/}$

$\boxed{F11=A_{\text{N}}-B+B_{\text{N}}-C_{\text{N}}D}$

$\boxed{\%}$

$\boxed{\text{PATH BOOLEFV, LEIRAS, BEOLVAS}}$

Eredmény:



2. Adatmegadás

a. $\boxed{\text{/HERT/F1,F2,F3,F4,F6,AND//}}$
 $\boxed{\text{PATH BOOLEFV,LEIRAS,MUVELET}}$

b. $\boxed{\text{vvvvvvF6=-A}}$

$$\boxed{\text{F4=B+C+-D}}$$

$$\boxed{\text{F3=-BxC-D}}$$

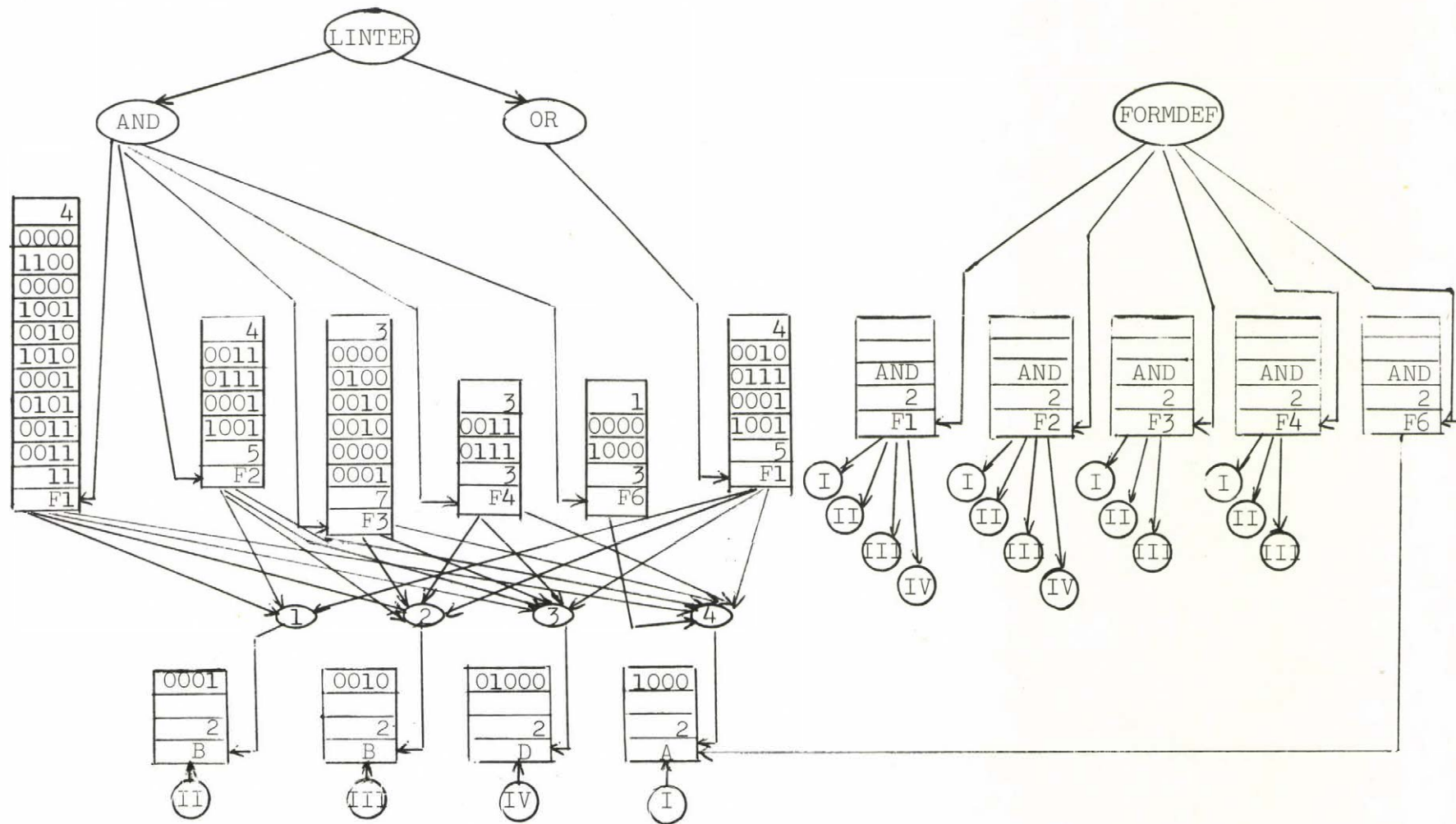
$$\boxed{\text{F2=-A+B/+/B+C+-D/}}$$

$$\boxed{\text{F1=-AxB+-BxC-D}}$$

$\boxed{\text{\%}}$

$\boxed{\text{PATH BOOLEFV,LEIRAS,BEOLVAS}}$

Eredmény:



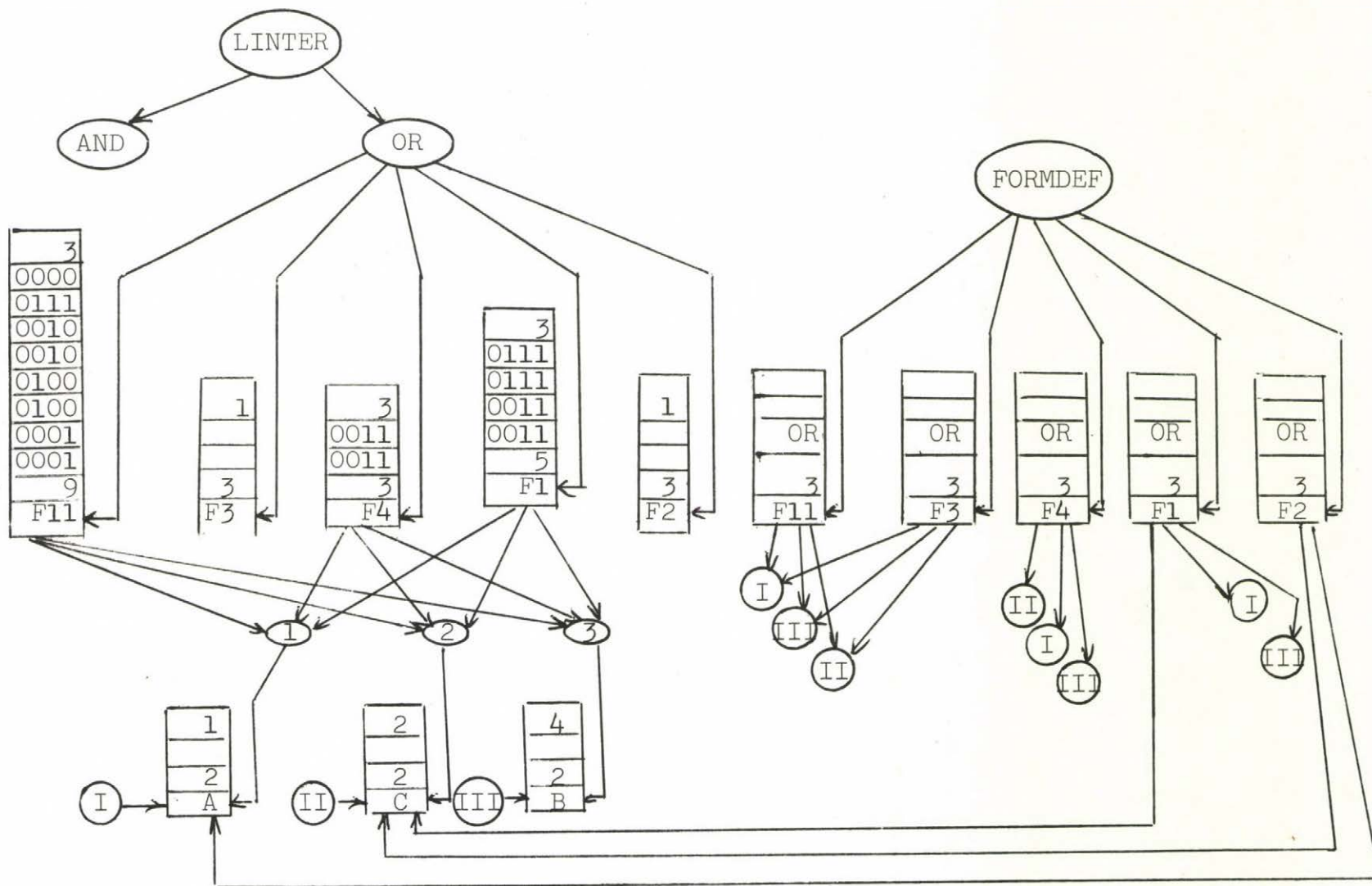
3. Adatmegadás

a. $\overbrace{\text{/HERT/F11,F3,F4,F1,F2,OR//}}$
 $\overbrace{\text{PATH BOOLEFV,LEIRAS,MUVELET}}$

b. $\overbrace{\text{vvvvv F2=C\#/A\#-A/}}$
 $\overbrace{\text{F1=C\#/A+A\#B/}}$
 $\overbrace{\text{F4=C\#/A\#/A+B//}}$
 $\overbrace{\text{F3=/A\#B\#C/\#-/A\#B\#C/}}$
 $\overbrace{\text{F11=A+B+C+-/A+B+C/}}$

$\overbrace{\text{\%}}$
 $\overbrace{\text{PATH BOOLEFV,LEIRAS,BEOLVAS}}$

Eredmény:



4. Adatmegadás:

a.

/HERT/FZ, AND//

PATH BOOLEFV, LEIRAS, MUVELET

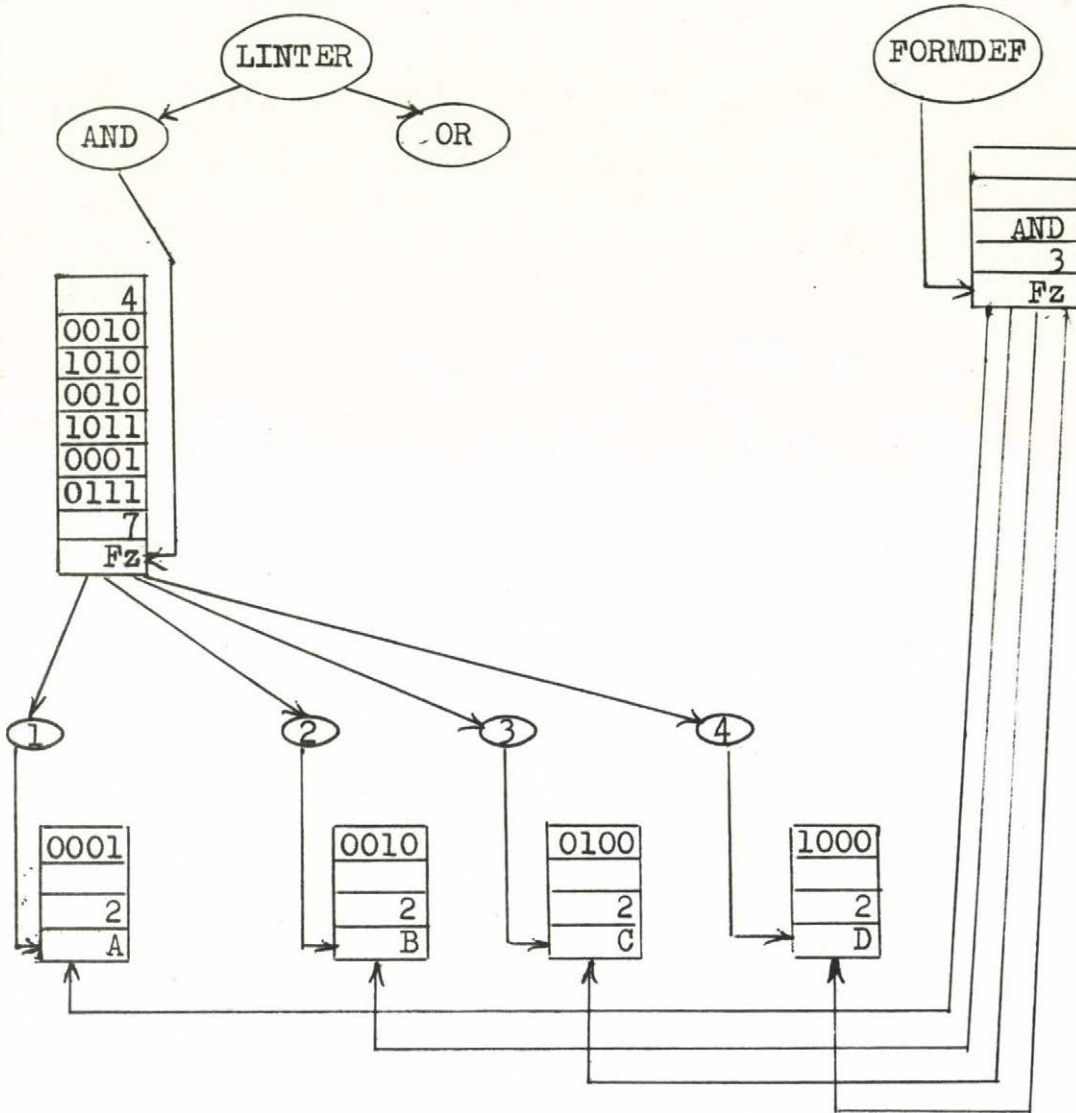
b.

vvvvvvv FZ=/A+-B+-C/≠/-A+B+-D/≠/B+-D/

%

PATH BOOLEFV, LEIRAS, BEOLVAS

Eredmény:

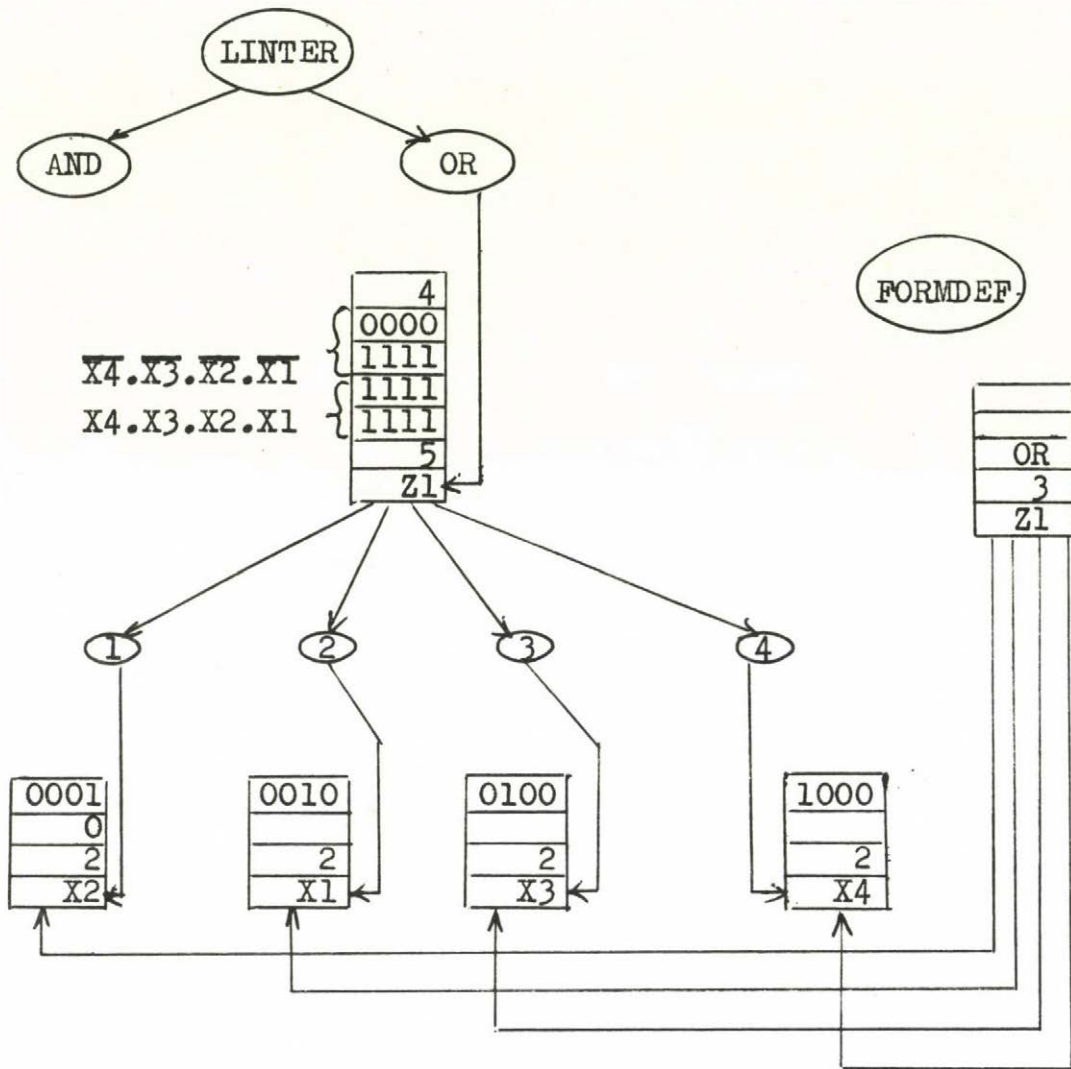


5. Adatmegadás

a. /HERT/Z1,OR//
PATH BOOLEFV,LEIRAS,MUVELET

b. A=NOR/X2,X3/
Z1=NOR/NOR/X2,NOR/X1,X3//,NOR/X1,A/,NOR/A,X4/,NOR/X3,
NOR/X2,X4///
%
PATH BOOLEFV,LEIRAS,BEOLVAS

Eredmény:



6. Adatmegadás

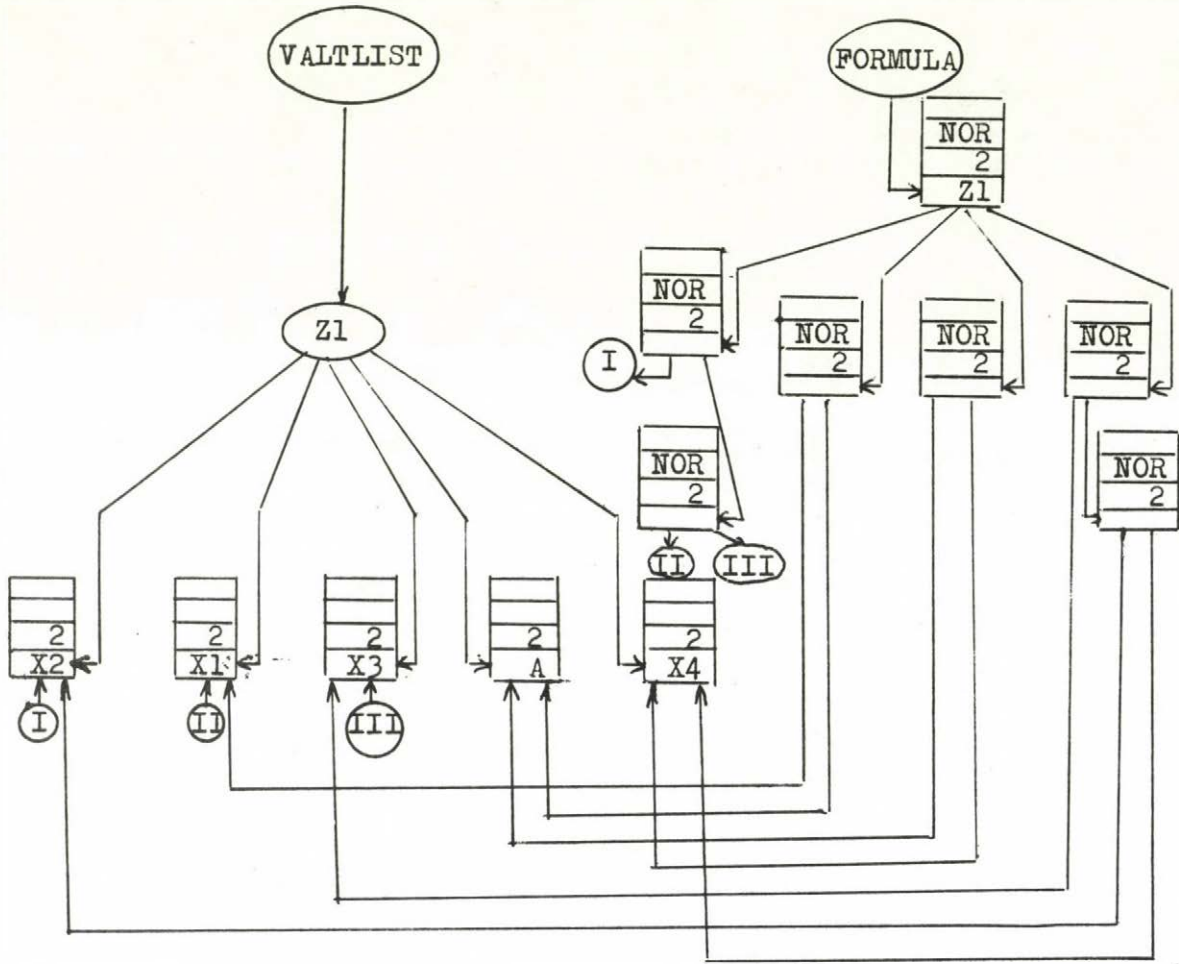
a. CHELY/UJZ1/Z1,X1,EGY//

PATH BOOLEFV,LEIRAS,MUVELET

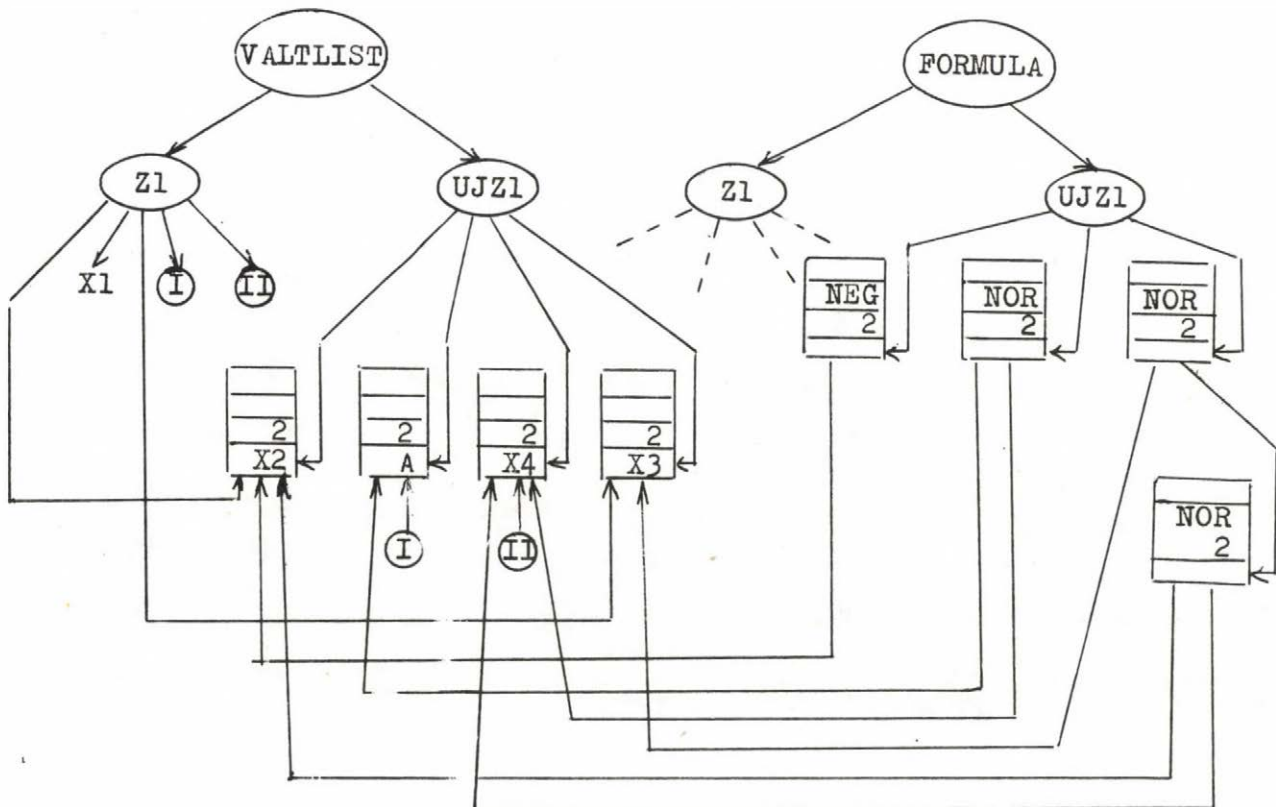
b. Azonos az 5.b-vel.

Eredmény:

a./ Lista állapota a változólista létrehozása után.



b./ Lista állapota a CHELY művelet végrehajtása után.



IRODALOMJEGYZÉK

1. Логический язык для представления алгоритмов синтеза релейных устройств /ред.: М.А.Гаврилов/
Изд.Наука, Москва 1966.
2. Fidrich Ilona, Uzsoky Miklós: LIDI-72
MTA SzTAKI Tanulmányok 16/1974
3. B. Randell; I. Russel: Algol 60 implementation.
Acad. Press. 1964.
4. Pásztorné Varga Katalin: Módszerek Boole-függvények minimális vagy nem redundáns, $\{\wedge, \vee, \neg\}$ vagy $\{\text{NOR}\}$ vagy $\{\text{NAND}\}$ bázisbeli, zárójeles vagy zárójel nélküli formuláinak előállítására.
MTA SzTAKI Tanulmányok 1/1973.

A T A N U L M Á N Y O K sorozatban 1978-ban megjelentek:

- 74/1978 Vorträge über das graphische Display GD'71
- 75/1978 Vaskövi István - Galbavy Márta: Anyagszétválasztási rendszerek tervezésének és optimális üzemeltetésének általános megközelítése
- 76/1978 Somló János - Nagy Judit: Módszer munkadarabok forgácsoló megmunkálási folyamatának optimalizálására.
- 77/1978 Szászné Turchányi Piroska: Optimalizálási feladatok csomagkapcsolt számítógéphálózatok tervezésénél
- 78/1978 Darvas Péter - Gallai István - Hosszu Péter - Krammer Gergely: Papers on Computer Graphics
- 79/1978 Dr. Adolf Kotzauer:
Beschriftung und Bemassung von automatisch erstellten Zeichnungen unter Benutzung des graphischen dialogs
- 80/1978 Studies in Applied Stochastic Programming I.
- 81/1978 Peter Bonitz: Ein Beitrag zur Theorie des Entwurfs doppelt gekrümmter Flächen unter differentialgeometrischen und rechentechnischen Aspekten.
- 82/1978 Tankó József: Szabályos job-folyam párok ütemezésének vizsgálata I.
- 83/1978 Tankó József: Szabályos job-folyam párok ütemezésének vizsgálata II.
- 84/1978 Bányász Csilla - Keviczky László: Discrete Time Identification of Linear Dynamic Process
- 85/1978 Dr. Hoffmann Péter: Számítógépes szerszámgepvezérlés egy alkatrészprogramozási módszere

- 86/1978 Ruda Mihály: A SIS77 statisztikai információs rendszer kialakításának szempontjai, alkalmazásának és továbbfejlesztésének lehetőségei
- 87/1978 Téli iskola - Operációs rendszerek elmélete

A T A N U L M Á N Y O K sorozatban 1979-ben megjelentek:

- 88/1979 Renner Gábor - Gaál Balázs - Hermann Gyula - Horváth László - Várady Tamás: Szoborszerű felületek tervezése és megmunkálása
- 89/1979 Ruda Mihály: A SIS77 statisztikai információs rendszer /a felhasznált számítástechnikai eszközök, a rendszer szerkezete és programjai/
- 90/1979 Bányász Csilla - Keviczky László: Optimum Insensitivity of the Linear-continuous Transformation
- 91/1979 Téli iskola /Szentendre/
- 92/1979 Bolla M., Csáki P., Fischer J., Herodek S., Hoffmann Gy., Kutas T., Telegdi L., Wittman I.: A balatoni ökoszisztéma modellezése
- 93/1979 Andor László: Kisgépes adatbázis kezelő rendszer
- 94/1979 Gertler János: Egy statisztikus szűrési eljárás számítógépes folyamatirányításához
- 95/1979 Báthory M.; Galló V.; Kovács E.; Mérő L.; Siegler A.; Vajta L.: Festőrobot vezérlésére alkalmas alakfelismerési berendezés
- 96/1979 Mérő László: Konturkeresés zajos digitalizált képekben



