POLYGON
VII. kötet 1. szám 1997.

Kalmár László, a magyarországi számítástudomány atyja

Varga Antal

40 éve Szegeden indult a hazai programtervező matematikus képzés

Kalmár Lászlónak, a zseniális matematikusnak nagyon sokszor jutott osztályrészül a küzdelem, a harc egy-egy tudományterület hazai meghonosításáért. Vagy talán pontosabb lenne azt mondani, sokszor vállalta a küzdelmet. Így volt ez a huszas évek végétől a matematikai logika érdekében, de így adódott az ötvenes évek közepétől kezdve, mikor is a kibernetikáért, a számítástechnikáért szállt ringbe. Az eredmények fényesen igazolják, megérte.

A továbbiakban a számítástechnikáért, a számítástudományért folytatott utolsó nagy küzdelméről kívánunk szólni.

A számítástechnika élesen elhatárolható két korszakra osztható: a Neumann előttire és a Neumann utánira. Neumann-nak, az univerzális zseninek, prófétai meglátásai minőségi változást hoztak e területen. Persze nem feledkezünk meg arról, hogy a gondolatokat sok-sok ember kollektív fáradozása csíráztatja és érleli meg, de Neumann, a géniusz nemcsak a formát, a minőséget adta hozzá. Amit ő akkor lefektetett, ma is az alapot jelenti.

A. M. Turing angol matematikus, a matematikai logika neves művelője, aki doktori disszertációját a princetoni egyetemen készítette, Neumann-nal is kapcsolatban volt. Egy 1936-ban megjelent dolgozatban megmutatta, hogy azzal a géppel, amely képes arra, hogy elvégezzen néhány alapvető műveletet, minden olyan számítás elvégezhető, amelyhez létezik algoritmus. Turing gépe persze nem volt fizikai értelemben gép. Neumann, Turing eredményei alapján, arra az álláspontra jutott, hogy nem speciális gépeket kell építeni, hanem univerzális gépeket, melyek minden olyan művelet végrehajtására programozhatók, amely műveleteket egyértelmű szabályokba lehet foglalni. Vagyis a számítógépek „univerzális Turing-gépek" legyenek. Ezek persze az alkalmazási lehetőségeket is nagy mértékben kiszélesítették.

Ha Neumann korszakalkotó javaslatait pontokba kívánjuk szedni, akkor ezek a következők lehetnének:

1. *A gépnek teljesen elektronikusnak kell lenni.*

Ez a mechanikus alkatrészek kiküszöbölését célozta, így a biztonság, az élettartam, a gyorsaság fokozására nyílik lehetőség.

2. *Belső memóriát kell létrehozni.*

Ez nyújt lehetőséget arra, hogy az utasításokkal is műveleteket hajthassunk végre, ami megszünteti azt a szűk keresztmetszetet, ami a műveletek programozását (a külső programozást, az emberi kéz által végrehajtandó átdugaszolást) jelentette. Ez a programok automatikus végrehajtását tette lehetővé.

Így lehetővé vált a 3. javaslat megvalósítása: *A tárolt program.*
A belső memóriában az adatoktól jól elhatárolva kerüljön elhelyezésre.

4. *Az univerzális számítógép megalkotása.*
Erről fent szóltunk.

5. *A kettes számrendszer alkalmazása.*
Neumann és éppúgy Kalmár is a matematikai logika kiemelkedő művelői voltak. Ebből adódóan a számítástechnikát, a számítástudományt úgy tekintették, mint a matematika részét, másképpen látták ezt a „világot”. A tudomány alapjainak és eszközeinek a matematikai logikát tartották. Elsősorban ehhez volt szükség arra, hogy kettes számrendszer legyen a gépen megvalósított számrendszer. Lehet, hogy Neumann ellenőrzésképpen megvizsgálta, melyik alapszám lenne az optimális a számábrázolásra, erről is lehet olvasni. Sokan a technikai megvalósítás okán hiszik a kettes számrendszer javaslatát. Egyik sem zárható ki, de a Neumann-alapelvek lefektetésénél a megvalósítás még hátra volt, a technikai nehézségek még nem jelentkezhettek, a gép építése még el sem kezdődött.

Neumann javaslatai közel 50 éve hangzottak el, az akkori elektronikus alkatrészek reménytelenül elavultak. Még Neumann sem tudta megbecsülni azt a fejlődést, ami bekövetkezett, a számítógépek óriási térhódítását, kapacitásának hihetetlen megnövekedését. De az általa lefektetett, fentebb felsorolt elvek ma is hiánytalanul érvényesek, és jelentik az automatikus univerzális elektronikus számítógépek alapelveit.

Hogy mindezek Neumannhoz kapcsolhatók, igazolják a „szemtanúk”, a közvetlen munkatársak, többek között H. H. Goldstine, J. G. Kemeny visszaemlékezései, írásai. Ennyit el kellett itt is mondanunk, mert az utóbbi időben, különösen Neumann választott hazájában fellelhető megnyilvánulások, akarva akaratlanul, de szívesen megfeledkeznek erről.

Elmondhatjuk, Kalmár László személyében itthon is volt egy Neumannunk, aki nem aposztrofálható csak úgy, mint egyike a számítástechnikusoknak.

A matematikai logika el nem ismerése szakmai körökben még az ötvenes években is tartott, állítván, hogy e tárgykörnek soha nem lehet semmi alkalmazási területe, annyira elméleti, hogy az még a matematikusoknak is sok. A még „fiatalabb” kibernetikát egyenesen burzsoá áltudománynak minősítették, melynek célja a szocialista világ félrevezetése, felpuhítása. N. Wiener, Cybernetics könyvét betiltották. Függetlenül a tényektől, hogy Shannon amerikai mérnök, Sesztakov szovjet fizikus mérnöki alkalmazások sorát mutatta be, fedezte fel.

Addig csupán elméleti kérdésekkel foglalkozó univerzális tudósnak az első inpulzust egy 1955-ös drezdai kongresszus adta meg, mely a „Korszerű számítógépekkel kapcsolatos kérdések” címmel lett meghirdetve. Kalmár azonnal átlátta a téma rendkívüli jelentőségét. A téma elkötelezettje lett, és hazatérte után azonnal munkához látott.

Kalmár első törekvéseit is sokszor próbálták leállítani, de mindig sikerült kivédenie egy-egy hasonló területen tevékenykedő szovjet matematikus nevének felemlítésével és eredményeinek bemutatásával.

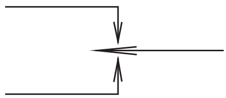
Így, nem törődve a gáncsoskodásokkal, a gáncsoskodókkal, 1956. április 10-én beindította kibernetikai szemináriumát 10 fővel. Tagjai mérnökök és matematikusok voltak. A lelkes csoportot időnként külsősként Tarján Rezső is segítette. Igazi úttörő munka indult, mert igaz ugyan, hogy a Neumann—Golstine-jelentés 1948-ban megszületett, de hétpecsétes titokként őrizték, ami miatt Kalmárék csak a Neumann előtti korszak eredményeiből, tapasztalataiból indulhattak el.

A szeminárium induló témájaként a matematikai logika műszaki és egyéb alkalmazásainak megismerését tűzték ki célul. Ugyanis Kalmár úgy képzelte, a matematikai logika csak sok lépcsőn keresztül tér vissza a valósághoz. A matematikára alkalmazható, mert szilárdabban megalapozott elméleteket lehet segítségével alkotni. Aztán ez a matematika hat az elméleti fizikára, a kísérleti fizikára, a technikára és így lehet belőle valami gyakorlati alkalmazás. Az ekkortájt megjelent Matematika alapjai egyetemi jegyzetében az ítéletkalkulus műveletei közül például az alternatív tagadás műveletéről vagy az együttes tagadás műveletéről úgy szól, mint elméleti kuriózumról. Nem hitte senki akkor még, hogy ezek lesznek — alig tíz év leforgása alatt — a többszintű logikai hálózat tervezésének eszközei.

Később a szemináriumon egy kis elektronikus számítógép megépítésének gondolata merült fel, hogy legyen meg az absztrahálási alap, hogy legyen miből absztrahálni. Tarján lebeszélte a csoportot (ez még „toronyóra volt aranylánccal"). Inkább egy logikai gép megépítését ajánlotta a tapasztalatok megszerzésére. Ebben az esetben ugyanis duplán kell „használni" a matematikai logikát, hiszen a probléma is logikai, meg az is, amivel meg lehet oldani. Így ilyen irányt vett a felkészülés. A kivitelezés az őszi, októberi események miatt akadozott, mivel a szükséges anyagok beszerzése sokszor lehetetlenné vált. Ezért a megépített gépet csak 1958. május elején tudták bemutatni. A munkát az is vezérelte, hogy olyat kezdtek el, amit más nem csinált Magyarországon, logikai gépet építettek. Az áttanulmányozott irodalomból kiderült, hogy az eddigi logikai gépek nem jók semmire, pl. nincs memóriájuk stb.

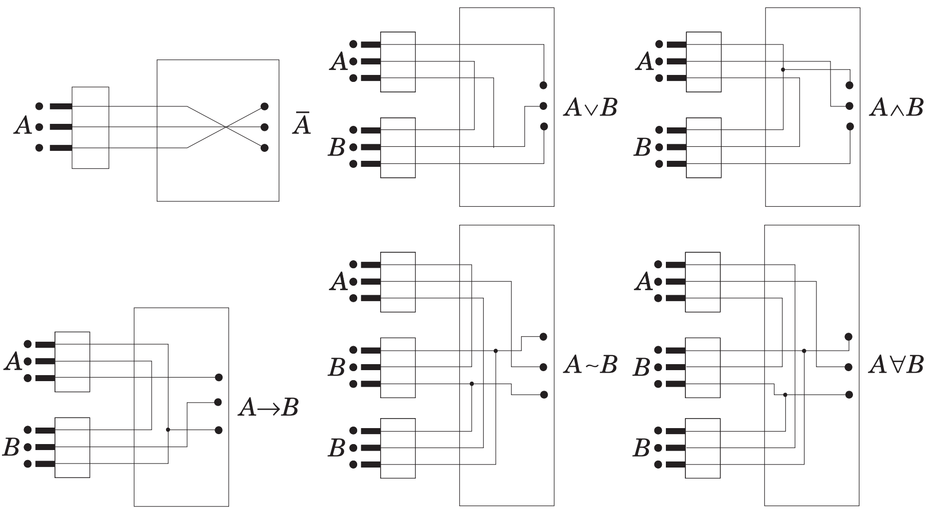
A terv egy digitális automata (logikai gép) megépítése lett, mely kétértékű logikai változókból, a konjunkció, diszjunkció, implikáció, ekvivalencia, kizáró diszjunkció, negáció akárhányszori alkalmazásával felépített formuláról meg tudja állapítani, hogy a logikai változók mely értékeloszlásainál „igaz", melyeknél „hamis" az értéke. Már a tervezés alatt kiderült, hogy a logikai gép alkalmas gyakorlati feladatok ellátására, az elektromechanikus áramkörök tesztelésére (tranzisztoros áramköröket még nem építettek ekkor).

A bemenet, a logikai formulák bevitele a gépbe dugaszolással történt. Ehhez Kalmár egy nagyon elmés és roppant elegáns megoldást talált. Kalmár gépe végtelenül egyszerű, ellentétben a relés, elektroncsöves, később a tranzisztoros megoldásokkal. Az ő gépe csak huzalokból állt. A huzal egyik végén egy hüvely volt, a másik végén egy dugó. Míg a szokásos eljárásoknál egy változó logikai értékét egy zárt vagy nyitott kontaktus ábrázolja, addig Kalmár ún. változókontaktust használt (1. ábra).



1. ábra

A változó kontaktus egy kettős érintkező: a középső vezeték vagy az alsó, vagy pedig a felső érintkezővel érintkezik. Ha a logikai változó igaz, a középső huzal például az alsó huzallal van összekötve, ha pedig hamis, a felső huzallal van összeköttetésben. Kalmár 3 huzalos megvalósítása redundáns megvalósítás, de ez a redundancia valójában előny, mert egyúttal hibajelzést is szolgáltat. Ezzel szemben hátrány az, hogy ha ugyanarra a logikai változóra több dobozzal kell rácsatlakozni, nem ágazhatunk le párhuzamosan, hanem mindannyiszor új érintkezőpárt kell beiktatni. A 2. ábra a negáció (-), a diszjunkció ($\vee$), a konjunkció ($\wedge$), implikáció ($\rightarrow$), kizáró diszjunkció ($\forall$) és az ekvivalencia ($\sim$) huzalos megvalósítását szemlélteti.



2. ábra

A huzalozott bemeneti egységek a bemeneten kívül a digitális számítógépek aritmetikai egységének megfelelő logikai egységek szerepét is betöltötték. Az adott logikai formulák közül az „igaz” értékűekre vonatkozó ítéletek bevitele a gépbe külön e célra készített, ugyancsak (kapcsolókon és jelzőlámpákon kívül) kizárólag huzalozott egység segítségével történt. Ez a doboz másik állásba átkapcsolva olyan feladatok megoldására tette képessé a gépet, hogy számlálja meg, adott logikai formulák közül hánynak igaz az értéke.

A gép *vezérlése* elektromechanikus volt.

A gép *memóriája* jelfogós, 8 bites, de a dugaszolás útján felépített huzalos áramkör is szolgált memóriacélokat, amennyiben magát a vizsgálandó formulát tárolta a vizsgálat befejezésig. Ezért a memóriát csak többütemű áramkörök vizsgálata esetén vették igénybe.

A gép *kimenete* jelzőlámpákkal volt megoldva, ezek mutatták a gép állapotát, a logikai változók és a vizsgált formula — többütemű áramkörök vizsgálata esetén formulák — logikai értékét. Továbbá jelezték, hogy a vizsgált formulák közül hánynak igaz a logikai értéke.

A gép *kapacitása* legfeljebb 8 logikai változót tartalmazó, tetszőleges bonyolultan felépített logikai formula vizsgálatát tette lehetővé. Többütemű áramkörök vizsgálata esetén a bemenő és közvetítő egységek száma együttvéve 8, a közvetítő és kimenő egységek száma együttvéve 12 lehetett.

A gép aktív része 52 jelfogó.

A logikai gépről azért szólunk kissé részletesebben és hosszabban, mivel a közben született gondolatok a későbbiekben is megtermékenyítőnek és meglepően előremutatónak bizonyultak. A logikai gép megépítése egyrészt igen jó beletanulást jelentett az elektronikus számítógépek megvalósításába, másrészt az a tapasztalat, hogy minden aritmetikai műveletet vissza lehet vezetni logikai műveletekre, elvi lehetőséget nyújtott arra, hogy lehet olyan számítógépet építeni, amelyben az aritmetikai egység kizárólag huzalokat tartalmaz. Ez abszolút gyors áramsebességgel dolgozik, ugyanis nincs késleltetés, igaz a változók számával exponenciálisan növekszik a huzalok száma, azaz csak úgy bővíthető, ahogy egy NP teljes algoritmus építkezik.

De előny is felfedezhető, pl. ez a megoldás egy jobb közelítést adó agymodellezésre adhat lehetőséget. Ha a huzalokat idegpályákkal helyettesítjük, akkor egy ma elképzelt agyi struktúra egyik közelítő modellje ismerhető fel.

Másrészt a logikai gépet dugaszolások segítségével lehetett programozni, amiről rögtön kiderült, hogy nehézkes és komoly hibaforrás. Ezért került sor egy billentyűs berendezés megalkotására, amely a logikai formula alapján felépítette a megfelelő áramkört. Ez egyenesen sugallta azt, hogy logikai formula helyett valamilyen programozási nyelven írt programnak a jeleit kell sorra bebillentyűzni, így egy fordítóprogram nélküli elektronikus számítógép megépítésének körvonalai rajzolódtak ki. Ezzel megszületett a formulavezérlésű számítógép elve, amely egyúttal egy generációs ugrást is jelentett. Ezután minden lépés, kísérlet közvetve vagy közvetlenül egy új elven épülő számítógép megalkotását szolgálta.

1959. szeptember közepén készült el a kettes számrendszerben működő huzalos összeadó. Feladata két, a kettes számrendszerben megadott természetes szám összeadásának bemutatása. A huzalos összeadó illeszthető lett volna az akkori legmagasabb színvonalon álló elektronikus számítógéphez. A Kalmár által tervezett és munkatársai által megépített egység az áram terjedési sebességével működött úgy, hogy minden, az ily módon megépített egységben az utasítás végrehajtása annyi időt igényelt, mint amennyi a memória hozzáférési ideje.

Vezérlése is tisztán huzalos. A bemutató modellnek memóriája nem volt. Kimenete 1-1 jelzőlámpa. Kapacitása legfeljebb 12 jegyű összeadandók kettes számrendszerben. Aktív alkatrészei nem voltak. A „tisztán huzalos” egységek megoldásának elvét a 147.135 sz. alatt magyar szabadalom védte.

A logikai géphez egy automatikus formulaközlő terve is elkészült, kivitelezését az 1962. évben jelölték meg. Célja a hibaforrások csökkentése. Kapacitása legfeljebb 8 változót és 12 logikai jelet tartalmazó formula billentyűs bevitelére adott lehetőséget.

Az ötvenes évek második felére egyértelművé vált, hogy a digitális elektronikus számítógépek alkalmazásának szűk keresztmetszete a programozás. Ellentétben az érdemi számítással, amelynek volumene akkor legfeljebb néhány óráig szokott tartani, a programozás napokig, hetekig, sok esetben hónapokig tartott. Eredménye, az ún. gépi program áttekinthetetlen, így a benne rendszerint előforduló hibák megkeresése és kijavítása szintén fáradságos munka volt.

Ezen a nehézségen világszerte az automatikus programozás módszerével igyekeztek segíteni. Ehhez fordítóprogramok megírására volt szükség. Persze itt is alapvető problémák merültek fel.

A fordítóprogramok a számítógépek típusától függően néhány ezertől tízezerig terjedő utasításból álltak. Megvalósításuk tekintélyes számú magasan kvalifikált matematikus szakember többhónapos munkáját kívánta meg, majd nem kevés időt vett igénybe a hibák felderítése és kijavítása is. Sok helyen inkább a gépi programozás fáradságosabb munkáját választották, mivel sok esetben egy-egy fordítóprogram elkészülésekor már aktualitását veszítette, mivel közben maga a számítógép modernizálódott. Egyes számítógépgyártók gépeiket fordítóprogrammal hirdették, valójában azonban legtöbb esetben ún. értelmező programokat szállítottak, amely értelmező jelentősen meghosszabbította a gépek futásidejét.

Megjegyezzük, hogy a 70-es években, mind a fordítóprogramok elméletében, mind pedig a fordítóprogramok gyakorlatában ugrásszerű fejlődés következett be, így a fent említett hiányosságok mára már többségükben elmúltak.

Így nem meglepő, hogy több helyen felvetődött a probléma más megoldásának gondolata. Mégpedig olyan digitális számítógép megvalósítása, melyet a gépi program, a gép„anyanyelve” helyett azok a formulák vezérelnek, amelyek együttesen — a matematika, a matematikai logika és a számítástechnika szokásos jelöléseivel — kifejezik az elvégzendő számítás algoritmusát.

Így egymástól függetlenül F. L. Bauer és K. Samuelson (München, Mainz, 1957.), W. Kammerer (Jena, Berlin, 1957.), Kalmár (Szeged, 1959.) kezdtek el a formulavezérlésű számítógépek tervével foglalkozni. Később W. Pawlak (Warsó, 1960.) és V. M. Gluskov (Kijev, 1963.) csatlakoztak a témához.

Bauer és Samuelson 1960-ban megjelent publikációjából kiderült, hogy elképzelésük műszakilag lényegesen bonyolultabb és költségesebb gépet jelent Kalmárénál. Kammerer gépe is költségesebb volt Kalmár gépénél, de műszaki megoldása közelebb állt hozzá. Pawlak gépe egy célgép volt, nem univerzális, függvénytáblák összeállítására készült a Varsói Műszaki Egyetemen. Gluskov gépe az ALGOL-60 nyelven felírt utasításokkal vezérelt gép volt, elvében a Kalmár-gép ismerhető fel. Az akkortájt gyakran Szegedre látogató Gluskov munkatársak Kalmártól tanulhatták meg a formulavezérlésű számítógépek csínját-bínját. Az egységek neveire is Kalmár segített orosz megfelelő neveket megalkotni, illetve keresni.

Kalmár tervét először Varsóban adta elő 1958-ban. A gép felépítését, műszaki leírását és működését a függelékben adtuk meg.

A történések megértéséhez néhány szót kell szólnunk az „adminisztratív történésekről" is. Az 1956. április 10-én beindított kibernetikai szeminárium tevékenysége és eredményei, továbbá Kalmár kitartó küzdelme révén (pontosan egy évvel később) létrehozták (Szegeden) a Magyar Tudományos Akadémia (MTA) Matematikai Kutató Intézet „Matematikai logika és alkalmazásai" csoportját, amely 1958-tól 1967-ig mint osztály működött (tudományos vezetője és irányítója Kalmár László). 1967-től a — Matematikai Kutató Intézetből kiválva - az MTA „Matematikai logikai és Automataelméleti Tanszéki Kutató Csoport" lett. 1963-tól megalakult az egyetem Kibernetikai Laboratóriuma, melynek tudományos vezetője szintén Kalmár László lett.

1963-ban elkészült a formulavezérlésű számítógép terve, pénzügyi kalkulációja. A gép anyag- és alkatrészköltsége 268 500 Ft volt. Személyi igény: 1 osztályvezető, 1 tudományos munkatárs, 1 villamosmérnök, 1 műszaki rajzoló, 3 szakmunkás, 2 adminisztrátor. Az összeg abban az időben felemésztette volna a Mat. Kutató éves keretének tekintélyes részét, éppen ezért Rényi Alfréd, az akkori intézetigazgató javasolta, hogy hozzanak létre egy szegedi matematikai kutatóintézetet. Kalmár problémáját ez nem oldotta volna meg, mivel helyiségbővítésre mód nem volt, így a meglévő adottságok inkább beszűkültek volna. Akkor olyan kísérleti gép megoldását javasolták, mely oktatási célokat szolgálna, és amellett kisebb volumenű számítások elvégzésére is alkalmas lenne. Erre sem jutott elegendő pénz. A próbálkozások 1966. március végéig húzódtak, majd az akadémia részéről Budó Ágoston, Pál Lénárd, Kónya Albert, Tarján Imre fizikusok és Hajós György matematikus meglátogatták a matematikai logika és alkalmazásai osztályt és azt javasolták, hogy a formulavezérlésű számítógépekkel kapcsolatos kutatásokkal kooperáljanak a kieviekkel, és folytassanak közös munkát az Ukrán Tudományos Akadémia Gluskov által vezetett Kibernetikai Intézetével. Kalmár ebbe beleegyezett, aminek következtében aztán Kalmár gépe kisebb-nagyobb módosításokkal, de az ő publikációiból és személyes előadásai, konzultációi révén megépült. Gluskovék a MIR nevet adták neki, a nyelve nem a matematika formula nyelve, hanem az ALGOL-60-hoz közel álló nyelv lett. Így legalább Kievben megvalósultak Kalmár legfontosabb elképzelései, ami alapján biztatást is kapott a későbbiekhez. A gép Magyarországon (többek között az örökös pénzhiány következtében) nem valószínű, hogy megépítésre került volna.

A formulavezérlésű számítógép egy újabb felvonása indult el 1973. március 1-vel. Az MTA Természettudományi Főosztálya részéről felkérés érkezett Kalmárhoz, vegyen részt az ESZR (a volt szocialista országok Egységes Számítógép Rendszerét hívták így) távlati fejlesztési programjában, és dolgozza ki a „Belső gépi nyelvek, beleértve a magasabb szintű nyelveket" című témát.

Erre egy 12 fős munkacsoport alakult, és Kalmár irányításával dolgozott. E téma harmadik fejezeteként került be a Kalmár-féle formulavezérlésű számítógép korszerű alakjának megtervezése. E téma specialistája Makay Árpád volt. Az első formulavezérlésű gép megvalósításának célja az lett volna, hogy ha már egyszer a számítógép belső nyelvén való programozás az általános, akkor legyen az a belső nyelv minél magasabb szintű, és így a gép működni tud fordítóprogram nélkül. A programot ugyanis egy magasabb szintű nyelven kellett megírni, ami egyúttal a gép „anyanyelve". Így az első formulavezérlésű gépet egy egynyelvű gépnek képzelték el. A hetvenes évekre már általánossá vált a magasabb szintű algoritmikus nyelveken történő programírás. Sőt, a különböző alkalmazásokra több ilyen algoritmikus nyelvet dolgoztak ki: ALGOL, FORTRAN, COBOL, PL-1 stb., hogy csak a legismertebbeket említsük. Emellett különböző szoftver és hardver segédeszközök alakultak ki, mint pl. a rendszerprogramíró nyelvek és ezek implementálása, a beépített stack, stb. Ennek következtében egy korszerű formulavezérlésű számítógépnek többnyelvűnek kell lenni, amelyben az alkalmazások szempontjából legfontosabb algoritmikus nyelvi elemek implementálva vannak. Aminek következtében a formulavezérlésű számítógép belső nyelve sokkal közelebb áll az algoritmikus forrásnyelvekhez, mint a szokásos számítógépek „anyanyelve". Ebből aztán következik az is, hogy a formulavezérlésű számítógép műszaki megvalósítása nagyobb ráfordítást kíván, mint egy akkori szokásos számítógép. Többek között nagyszámú regisztert igényelnek, továbbá rendelkeznie kell megfelelő, többszintű megszakítási rendszerrel és erre támaszkodva kidolgozott operációs rendszerrel. De éppen a nagyobb ráfordítások miatt nagyobb mértékű univerzalitás valósítható meg. Ennek eredményeképpen a korszerű formulavezérlésű számítógépek alkalmasak (tudományos, műszaki, adatfeldolgozási, gazdasági feladatok mellett) rendszer-programozási, folyamatirányítási, szöveg, lista, fa és általánosabb gráfkezelési stb. feladatok megoldására is.

Szükségesnek látszott (és ez később is igazolódott), hogy a számítógépeket és tulajdonságaikat matematikai eszközökkel írjuk le. Kalmár László sokat foglalkozott azzal a kérdéssel, hogy a matematika mely ága képes erre. Az algebra tűnt számára a legmegfelelőbbnek.

Az elsők között ismerte fel az Univerzális algebra szükségességét és adta meg alapvető fogalmát. Az 1955-ös balatonvilágosi algebrai konferencián a struktúra általános definíciójára világított rá. Ez a fogalom az algebrában és a matematikai logikában használatos struktúrafogalom egyesítését és egy új szemlélet elindulását jelentette. Ezen a konferencián felvázolta, milyen irányba halad az algebra. A fejlődés túlment az egyes speciális addig vizsgált struktúrafajtákon. Egy struktúra egy halmazból, műveletekből (amely műveleteket a halmazelemeken el lehet végezni), továbbá relációkból (amelyek vagy fennállnak, vagy nem állnak fenn az elemek között) épül fel. A műveletek eredménye a rendszernek, a struktúrának az eleme. A relációk nem egy újabb elemet szolgáltatnak, hanem logikai értéket. Kalmár felhívta az algebristák figyelmét arra, hogy ma már az algebra úgy kezeli a struktúra fogalmát, mint egy tetszőleges halmazt, tetszőleges műveletekkel és relációkkal meg tetszőleges axiómákkal. A feladat pedig az, hogy ilyen általánosságban érvényes tételeket bizonyítson be a struktúráról. Szele Tibor erről az előadásról nyilatkozott úgy, hogy Kalmár egész életre szóló tennivalót adott az algebristáknak. E kitérőre azért volt szükség, hogy az 1973-as folytatást megértsük. Kalmár később azt is felismerte, hogy az általa megfogalmazott struktúrafogalom (amellett, hogy úgy az algebrára, mint a matematikai logikára termékenyítőleg hatott) és a nyomán kialakuló univerzális algebra, mely felöleli többek között a halmazelméletet, még olyan kérdéseket is magával hozott, amelyek megválaszolására többek között a számítástechnikában van szükség. Ez rögtön felismerhető, mikor Kalmár azt mondja, hogy a modern formulavezérlésű számítógép belső nyelvének tartalmaznia kell azt, ami a különböző algoritmikus nyelvekben közös. Ez alapján a formulavezérlésű számítógéphez, belső nyelv egyaránt nagyon általános értelemben vett, elvileg akárhány típust megengedő formulanyelv lett megfogalmazva. Ennek szintaxisa független a számítógép alkalmazási területétől, de amely a különböző alkalmazási területek számára különböző szemantikával látható el. A gép megépítését a Központi Fizikai Kutató Intézet vállalta el. 1975-ben megkezdődtek az előmunkálatok, megkezdődött a felkészülés, a dokumentációk elkészítése. De Kalmár László 1976. augusztus 2-án bekövetkezett halálával mindezek abbamaradtak.

Kalmár számítástudományi elméleti kutatásai is igen nagy figyelmet keltettek. Az első hazai elméleti eredmények is nevéhez fűződnek. Az ALGOL-60 nyelv egy — a feltételes utasításokra vonatkozó — fogyatékosságára ő hívta fel az alkotók figyelmét, és javította ki az egyértelműséget sértő hibát.

Kalmár 1965-ben a véges automata fogalmának finomításával a számítógépnek egy algebrai modelljét vezette be, majd publikálta a modell egy egyszerűsített változatát. Számítógépen olyan

$$C = (R, r_0, E, I, X, Y, \varepsilon, n, S, \omega, \alpha) \qquad (*)$$

algebrai rendszert értünk, amelyben felsorolt 11 elem jelentése a következő:

(1) R olyan véges halmaz, amely legalább két elemet tartalmaz.

(2) r_0 az R halmaz egy kitüntetett eleme.

(3) E, I, X és Y véges, nem üres halmazok.

(4) n valamilyen természetes szám.

(5) ε a $E^I \times I \times X$ halmaznak az $A = R \times E^I \times I$ halmazba való olyan egyértelmű leképezése, hogy legfeljebb egy olyan $e(\in E^I)$, $i(\in I)$ és $x(\in X)$ elemhármas van, hogy az $\varepsilon \in (R \times E^I \times I)^{E^I \times I \times X}$ leképezés $\varepsilon_1 : E^I \times I \times X \to R$ első projekciójára $\varepsilon_1(e, i, x) \neq r_0$ teljesül.

(6) Az S halmaz $\sigma : I^n \to A^{A'} \times D$ alakú leképezéseknek egy rendszere, ahol $A' = R' \times E^I \times I$, $R' = R \backslash \langle r_0 \rangle$ és D az A' halmaznak az Y halmazba való összes (parciális) leképezéseinek halmaza.

(7) ω az E halmaznak egy-egy értelmű leképezése S-be.

(8) α az E halmaznak egy-egy értelmű leképezése I^n-be.

A C rendszernek mint számítógépnek a működését a következőképpen értelmezzük: Az R halmaz az *üzemmódok* reprezentálására szolgál. Feltételezzük, hogy C csupán az r_0 kitüntetett üzemmódban képes külső jeleket felfogni, az r_0-tól különböző R-beli üzemmódokban pedig automatikusan működik, külső beavatkozás nélkül végzi a számításokat, és szükség esetén kimenő jeleket bocsát ki.

Az E halmaz a *memóriarekeszek lehetséges tartalmainak* reprezentálására szolgál. (Az utasításszámláló nem tartozik a memóriarekeszek közé.) Továbbá feltesszük, hogy az E minden eleme lehet tartalma bármelyik memóriarekesznek.

Az I a *memóriarekesz indexeinek* halmaza, azaz a *címek* halmaza. Feltesszük, hogy a címek előfordulnak az utasításszámláló tartalmaként. Az n pozitív egész szám megadja, hogy a C hány címes gép. Így a lehetséges címek száma I^n.

Az X a *bemenő*, az Y a *kimenő jelek* halmaza.

Az $A = R \times E^I \times I$ halmazt a C számítógép *állapothalmazának* nevezzük. A C gép *állapotait* olyan (r, η, i) hármasokkal írjuk le, ahol $r \in R$ megadja, hogy a gép az adott időpillanatban milyen üzemmódban működik, az $\eta \in E^I$ hozzárendelés a címhez tartozó memóriarekesz tartalmát adja, az $i \in I$ pedig az utasításszámláló tartalmát. Egy $a = (r, \eta, i) \in A$ állapotot nyugalmi állapotnak illetve munkaállapotnak nevezzük aszerint, hogy $r = r_0$, vagy $r \neq r_0$. Vagyis a (6) alatt szereplő A' a számítógép munkaállapotainak a halmaza.

Tegyük fel, hogy a C számítógép egy $a = (r_0, \eta, i)$ nyugalmi állapotban van, és ebben az állapotban egy $x \in X$ bemenő jelet kap. Az a állapot most csak η-tól és i-től függ. Az η-hoz és i-hez, valamint x-hez az ε függvény egy $\varepsilon(\eta, i, x) \in A$ állapotot rendel. Azt mondjuk, hogy a C gép az a állapotból az x bemenő jel hatására az $\varepsilon(\eta, i, x)$ állapotba megy át. Feltételezzük, hogy C minden nyugalmi állapotában képes külső jelet felfogni.

A (σ, p) $(\sigma \in S,\ p \in I^n)$ párokat *utasításoknak* nevezzük. Ha a számítógép egy $a' \in A'$ munkaállapotban van, és egy (σ, p) utasítást kap, akkor C-nek egy parancsot kell végrehajtania, amelynek realizációja egy (δ, λ) függvénypárral adható meg, ahol $\delta : A' \to A,\ \lambda : B \to Y\ (B \subseteq A')$, és a δ függvénnyel leírható (az a'-től nem szükségképpen különböző) új állapotba megy át, és esetleg kiad egy $y \in Y$ kimenő jelet, amit a λ függvény ír le.

Kalmár professzor környezetében kialakult automataelméleti iskola is a hatvanas évek első felében kezdte meg ez irányú vizsgálatait.

Kalmár a matematikai nyelvészettel úgy került kapcsolatba, hogy az MTA Nyelvtudományi Intézete és az Eötvös Loránd Tudományegyetem Általános Nyelvészeti Tanszéke konferenciát rendezett a matematikai nyelvészet és a gépi fordítás kérdéseiről, és felkérték előadónak.

A matematikai nyelvészet kialakulását nagymértékben elősegítette, hogy az általános nyelvészek megismerkedtek a matematikának a strukturális viszonyokkal foglalkozó ágaival. Ez elősegítette a gépi fordítás követelményeinek kialakulását, lehetővé vált a természetes nyelveknek, nyelvi jelenségeknek axiomatikus és deduktív megközelítése. Lehetővé vált, hogy a természetes nyelvnek a közlési folyamatok beágyazásával — függetlenül a kommunikációs rendszerek jellegétől, a közlési csatornák szerkezetétől, a közlésben részt vevők sajátosságaitól stb. — a nyelvi jelenségeket sokkal szélesebb alapon vizsgálhassák. Ezen új nyelvtudományi ág egyik kidolozója N. A. Chomsky amerikai nyelvész és filozófus volt. Elméletét Chomsky-féle generatív grammatikaként emlegetik. Chomsky a természetes nyelveket maximális hosszúság nélküli, véges természetű rendszerként szemléli. Elméletének leglényegesebb tétele szerint véges számú szabályok segítségével helyes mondatok végtelen halmaza generálható (eltérően azon rendszerektől, melyek nem helyes mondatokat is generálnak). Kalmár, felismerve a Chomsky-féle generatív grammatikák rendkívüli jelentőségét, a 60-as évek elejétől behatóan foglalkozott a nyelvészet e modern szemléletű tudományágával, a matematikai nyelvészettel.

Ezen elmélet alapján a gépi hibakeresés is lehetővé vált, hisz a fent említett tétel következtében a szintaktikai hibák (a „nyelvtanilag, vagy mondattanilag helytelen mondtatok") felismerésére nyílt lehetőség. Kalmár 1964-től nyelvész és matematikus hallgatók részére szemináriumot tartott a témából. Megmutatta, hogy a nyelvészet és a matematika eredményei és módszerei hogyan alkalmazhatók hosszasan, kölcsönösen a két tudományban. Kezdetben, mint minden újonnan alakuló tudományágban, „egymás melletti elbeszélések", meg nem értésből fakadó viták is kikerekedtek. Ezek tisztázása révén néminemű átalakulások is létrejöttek. Így mára a *matematikai nyelvészeten* a nyelvészet egy matematikai eszközökkel operáló tudományágát értik. Ezzel szemben a nyelvészeti eszközöket használó, a matematikához tartozó tudományágat a *formális nyelvek (matematikai) elméletének* nevezik.

Kalmár a formális nyelvek definiálására új módszert, egy gráfmódszert dolgozott ki, melyet hatékonyan alkalmazott az oktatásban (lásd a függelékben).

Nem meglepő, hogy itt is, illetve ismét előkerül — Kalmár kedvenc vesszőparipája — a beszélt és a gépi nyelvek egy bonyolult, absztrakt algebrai struktúraként való kezelése. Kutatásaival a beszélt nyelvek elméletéhez is hozzájárult.

A formális nyelvek alapvető alkalmazási területe a programozási nyelvek lexikális és szintaktikai elemzése. Ez az elemzés nem rendel a programokhoz jelentést. A programozási nyelvhez a programok jelentését a szemantika definiálja, mint ahogy a beszélt nyelveknél a szöveg jelentését is.

Egy programozási nyelvet akkor nevezünk teljesen definiáltnak, ha adott a szintaxisa és szemantikája.

Kalmár (e témában) nagy figyelmet keltő eredményét adta elő 1967-ben egy budapesti szimpóziumon, ahol (matematikai logikai eszközökkel) bebizonyította, hogy egy program szemantikája — bizonyos szempontból — a program gépi kódra való fordításának szinonimája.

Az 1956-ban elindított szeminárium résztvevői közül egy ütőképes oktatógárda is kinevelődött. Ezután Kalmár elkezdte beadványokkal bombázni az Oktatási Minisztériumot, hogy engedjék meg, hogy a programozást értő szakembereket képezhessenek Szegeden. Természetesen elutasították. Egy kiskaput mégis talált: az egyszakos tanárképzés megszüntetésekor a minisztérium beleegyezett, hogy a kar dékánja a harmadéves tanárjelöltek 5%-ának megengedje két szakjuk egyikének elhagyását, a megmaradt szakjuk egy speciális területén elmélyültebb tanulmányok végzésének céljából. Így 1957 őszén elkezdődött a képzés három egyszakos (ahogy hallgatótársaik cukkolták őket, EDSAC-os) matematikus hallgatóval. (EDSAC volt a neve az első Európában épült elektronikus számítógépnek.)

Eleinte „táblaprogramozást” jelentett a programírás. Kalmár egy fiktív gépet definiált oktatási célokra (melyet aztán a számítástechnika fejlődése alapján folytonosan korszerűsített). Előtte nyilvánvaló volt, hogy a hallgatók programozási előadásain nem célszerű egy konkrét gépkonfigurációt kiválasztani, és mint etalonra, erre a gépre írni a különböző oktatóprogramokat, ugyanis ezek akarva-akaratlanul kihasználják az illető géptípus speciális adottságait. Ez egy példával is szemléltethető. Ha egy adott konkrét gépen egy konstanst utasításként értelmezve hajtunk végre, pontosan tudható, mi fog történni, és ez a tény a programok írásakor messzemenően kihasználható. Továbbá egy konkrét géptípus kiválasztása esetén meg nem válaszolható kérdést jelentett volna az is, hogy a kiválasztandó gép egy-, két- vagy háromcímes gép legyen.

Többek között ezek és persze számos más probléma indokolta egy fiktív gép megadását, melynek létezett egy-, két-, háromcímes változata is (1.1.01 és 1.2.1.01 mellékletek, lásd a függelékben). Ezek egyesítették az egyes géptípusok lényeges jellemzőit. A hallgatók, akik fiktív gépen tanultak programozni, konkrét gép mellé kerülve hamar felismerték az illető géptípus és a fiktív gép közös tulajdonságait, jellemzőit, és viszonylag gyorsan képesek voltak az elsajátított programozástechnikai módszerek alkalmazására a gyakorlatban is. Persze a fiktív gépen való programozásnak hátrányai is adódtak, többek között nem volt lehetőség a program gyakorlatban való kipróbálására, futtatására. Megfosztotta a hallgatókat a — didaktikai szempontból nem lebecsülendő — sikerélménytől. Továbbá lehetetlenné tette a visszacsatolást is.

Nem jelentett alapvető változást az sem, hogy — az Akadémián szovjet dokumentáció alapján, kisipari módszerekkel bütykölt — az Akadémián kiöregedett és a minisztériumnak ajándékozott M-3 számítógépet a minisztérium az egyetem Kibernetikai Laboratóriumának adományozta. Később, mikor az Országos Műszaki Fejlesztési Bizottság felfigyelt Kalmár és munkatársainak oktatási tevékenységére, eredményeire, egy szovjet gyártmányú Minszk-22 számítógépet szerzett számukra. Az akkori miniszter az ingyen (könyvjóváírással) juttatott gép átvételét azzal a feltétellel engedélyezte, hogy az egyetem semmiféle (személyi vagy dologi) fejlesztést nem kér, amíg a gép működik. Ez évről-évre növekvő munka vállalásával tette lehetővé újabb szakemberek képzését.

A „fejlesztési zárlatot” 1970-ben oldották fel, a miniszter „feje felett” hozott, a számítástechnika fejlesztésére vonatkozó kormányhatározattal. Meg is kérdezte Kalmárt akkor egy felelős minisztériumi tisztviselő, honnan tudta 1956-ban, hogy 1970-ben lesz egy ilyen kormányhatározat.

A Minszk-22-es gép beállítása után, tekintettel arra, hogy a fiktív gép a gépi kódú programozást szolgálta — igaz, nem lett kizárva egy assembly nyelv létrehozásának lehetősége sem — felmerült az ötlet, hogy jó lenne a Kalmár-féle fiktív gépet a Minszk-22-n szimulálni, ami aztán meg is valósult, és az 1970/71-es tanév első félévétől működött.

A Kalmár-féle gép definíciójában sem bemeneti, sem kimeneti egységek nem szerepeltek. Ennek oka az volt, hogy a gyakorlatban működő minden számítógép olyan szoftverrel rendelkezett, melyek az adatok ki- és bevitelét csaknem automatikusan végezték, ezekhez a felhasználók hozzá sem férhetnek, csak használhatják azokat.

Kalmár-féle fiktív gép alábbi típusai lettek szimulálva:

1) egycímes, indexregiszter nélküli gép,
2) egycímes, indexregiszteres gép,
3) kétcímes, indexregiszteres gép,
4) háromcímes, indexregiszteres gép.

A négy típus közös tulajdonságai a következők voltak:

a) a programok felírása tízes számrendszerben történt;
b) az operatív memória rekeszei 0-tól 1023-ig volt címezhető;
c) indexregiszteres gép esetén feltételezték, hogy a programozó számára elegendő indexrekesz áll rendelkezésre;
d) az indexregiszterek címrész hosszúságúak, és — a gyakorlattól eltérően — előjellel rendelkeztek;
e) a konstansok gépi ábrázolása középvesszős volt, vagyis a bináris vessző a rekesz belsejében volt;
f) az alfanumerikus jelek karakterenként voltak tárolva az egészrész vagy az első címrész végére.

Tekintettel arra, hogy nem beszélhettek fix rekeszhosszúságról, fel kellett tételezni, hogy a memóriarekeszek olyan hosszúak, hogy a tárolandó információ, akár konstans, akár utasítás volt, minden esetben elfért bennük.

Később, ahogy a számítógépek korszerűsödtek, a fiktív gépnek is mind korszerűbb fajtái kerültek be az oktatásba. Kalmár a programozás főkollégiumában a magasabb szintű nyelvek közül az ALGOL-60-at, majd emellett az ALGOL-68-at szerepeltette. A formális nyelvek definiálására adott gráfmódszerével oktatta az ALGOL-60, illetve az ALGOL-68 nyelvet is. Ez a módszer nemcsak a nyelvek jobb megértését, szerkezetének szemléltetését szolgálta, hanem az elkészült program ellenőrzését, illetve a hibát tartalmazó program hibáinak a megkeresését is megkönnyítette.

Kalmár ezeket az ábrákat az ALGOL-60 és az ALGOL-68 zászlós ábráinak nevezte. Az ALGOL-68 módjainak áttekintő és a módok metaprodukciós szabályainak zászlós ábráját a függelékben mutatjuk be (5. és 6. ábra). Megjegyzendő, hogy ma általánosan elterjedt a programozási nyelvek struktúrájának szemléltetésére ez a módszer különféle jelöléstechnikai variációkban.

A magyarországi számítástechnikában tevékenykedők jó része hosszú időn keresztül közvetlenül vagy közvetve Kalmár professzor tanítványai voltak. A számítógépek megjelenésekor jól felkészült szakembergárda állt rendelkezésre, ami nem kis mértékben szolgálta azt, hogy országunk elismertsége a számítástudományban az elsők közé emelkedett.

Sok mindenről nem szóltunk. Nem említettük az alakfelismerés területén végzett kutatásait, azon publikációit, előadásait, amelyekben a számítógépeknek a különböző tudományterületeken való alkalmazási lehetőségeire mutatott rá. Nem szóltunk a Kalmár-féle önreprodukáló automatatípusról, melynek a gyakorlati kivitelezése is megvalósíthatónak látszik, vagy a számítástechnika határterületeihez tartozó témák kutatásaiban elért eredményeiről. De úgy gondoljuk, Kalmár nagysága a fentiekből is világosan megítélhető.

Nem az a tudós volt, akinek fő ambíciója az lett volna, hogy minél több cikket írjon. Nagyon sok dolog van, ami nem cikk alakjában áll rendelkezésünkre, hanem levelek és kéziratok rejtik. Saját bevallása szerint is a tudományban nem volt hűséges. Mélyen szántó kalandozásainak két fix pontja volt: a matematikai logika és a számítástudomány. Ezeknek nemcsak kiváló művelője, hanem — hazánkban — meghonosítója is volt.

Egy matematikus géniusz, aki életének utolsó pillanatáig villámgyors volt, szerteágazó és szintetizáló. Mélyreható gondolataival újra és újra rabul tudta ejteni a kortárs óriásokat. Érdeklődő és kifelé néző volt, élete végéig befogadó elme maradt.

Azonnal átlátta minden új gép és rendszer programozási folyamatát. Kritikájával, meglátásaival csodálatra késztette a számítástudomány legismertebb művelőit. Nála az elméleti matematika és az alkalmazott matematika, a gyakorlat, nem jelszóban, hanem nagyszerű agyában párosult.

Körülötte némi kedves összevisszaságot, emberszeretetet és szellemi izgalmat érzett az ember. Küzdött nagyért és piciért, mindig használni és tenni akart, gyakran beleütközve az olcsóbb jólrendezettekbe, a mindig jólviselkedőkbe, a kimértekbe, az óvatosakba, akik nem látták, nem vették észre a közöttük járó igazi nagyságot. Ő volt az első a magyar számítástudományban, ő alapította a számítástechnikai oktatást hazánkban, és sokan vagyunk ebben az országban, akik azzal akarunk büszkélkedni, hogy a tanítványai voltunk.

Függelék

A Kalmár-gép felépítése és műszaki leírása nagy vonalakban

Nyelve: Ljapanov-féle operációs programozási nyelv algolizált (ALGOL-58) változata. A szokásos öt egység: *bemeneti* (B), *memória* (M), *vezérlő* (V), *aritmetikai* (A), *kimeneti egység* (K).

A szokásos kapcsolatok:

$B \to M$:	a program formulák alakjában (alapszimbólumai kódolva), adatok is: konstansok a formulákban decimális alakban.
$V \to M$:	az utasításszámláló tartalma a program egyetlen alapszimbólumát határozza meg.
$M \to V$:	ez a (kódolt) alapszimbólum átmegy az utasításregiszterbe.

$V \to M,\ V \to A$:	a vezérlő egység meghatározza, mit kell erre mint utasításra tennie a memóriának vagy az aritmetikai egységnek (valamely rekesz tartalmát áttenni valamely regiszterbe, vagy viszont, valamely műveletet végrehajtani stb.).
$A \to M,\ M \to A,\ A \to K$:	a teendő végrehajtása.
$A \to V$:	az utasításszámláló következő tartalmának beállítása.
$K \to A,\ A \to K$:	kinyomtatás előtti rekonverzió.

A bemenet fő egységei egy szalaglyukasztó és egy fotodiódás szalagolvasó. A szalaglyukasztó csak abban tért el a szokásostól, hogy billentyűzete a nyomdai szedőgéphez hasonlított. Ennek megfelelően több (attól függően, hányféle betűt és egyéb jeleket kívánunk a formulában használni, amikor is pl. x', x" az x-től különböző betűnek számítanak), 10—12 csatornás.

A memória szervezése egyrészt abban tér el a szokásostól, hogy külön rekeszei vannak az index nélküli betűkkel jelölt (skaláris) mennyiségek, és külön az indexes betűkkel jelölt tömbelemek (pl. a vektorkomponensek, mátrixelemek) tárolására, másrészt hogy az index nélküli betűvel jelölt skalár tárolására szolgáló rekesz címe mindig megegyezik a megfelelő betű bináris kódjával. Ezenkívül a memóriához tartozik az ún. tömbeligazító, amely rekeszeinek címei megegyeznek a tömbök jelölésére fenntartott betűk bináris kódjaival. E rekeszek a megfelelő tömb első elemének címéről és az egyes elemek egyes indexei 1-gyel való növeléséhez tartozó címlépésekről szóló (a gép által a tömbök indexhatárait megadó formulák alapján kiszámított) információ tárolására szolgálnak.

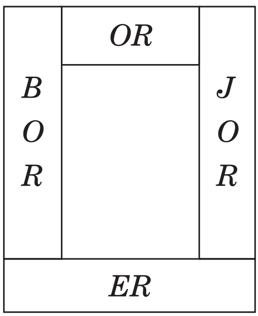
A vezérlő egység a szokásos *utasításszámlálón* és *utasításregiszteren* (az utóbbiba a számítási algoritmust kifejező formulák jeleinek kódjai kerülnek be egymás után) kívül tartalmaz egy ún. *aktiváló regisztert* az aritmetikai és a vezérlő egység bizonyos regiszterei közül annak (az ún. aktív regiszternek) a jelzőszámának tárolására, amelynek a formula következő jele által szolgáltatott információ feldolgozásában van szerepe. Továbbá tartalmaz egy ún. *szubaktiváló regisztert*, amely ideiglenesen átveszi tárolásra az aktiváló regiszter tartalmát mindaddig, amíg az az indexes mennyiségek indexeinek értékei kiszámítása során más célra van igénybe véve. Emellett tartalmaz a logikai formulák, címkék és ugrásjelek által adható ugrási utasítások, valamint az ugyanazon formulasorozattal megadott számítási eljárásnak többszöri ismétlésére, ugyancsak formulákkal adható utasítások feldolgozásához szükséges regisztereket.

Az *aritmetikai egység* az alapműveletek elvégzésére szolgáló szokásos áramkörökön kívül tartalmaz: *behuzalozott szubrutinokat* (pl. exp, ln, sin, cos, tg, ctg, arcsin, hatványozás, gyökvonás stb.) az elemi függvények számára. (Itt a behuzalozás persze nem a fent említetteket jelenti, hanem az illető utasítás kódjára egy alprogram hajtódik végre a gépben). Tartalmaz még *számátalakító regisztereket* a formulákban szereplő, decimális számrendszerben felírt konstansoknak gépi (bináris számrendszerbeli) alakra való átalakításához, *címkiszámító regisztereket* az indexes betűkkel jelölt tömbelemek címeinek indexeik, valamint a tömbeligazító megfelelő rekeszében tárolt információ alapján történő kiszámításához, végül *formulafeldolgozó* regisztereket az aritmetikai és a logikai formulák egyes jelei szolgáltatta információ feldolgozására.

Az aritmetikai egység többi részei működésük során természetesen felhasználták az alapműveleteket végző áramköröket és a regisztereket is.

A *kimenet* fő egységei: egy (vagy önálló, vagy az aritmetikai egység alapműveleteket végző áramköreit és regisztereit felhasználó) számátalakító a kinyomtatandó számítási eredményeknek binárisból a decimális számrendszerbe való visszaalakítására, és egy, a szám-visszaalakító, ún. *nyomtató regiszter* által vezérelt géptávíró.

Aritmetikai kifejezések feldolgozása. Hogy a gép működését jobban megértsük, mindenekelőtt a (Kalmár-féle elgondolás szerinti) formulavezérlésű számítógép *„kifejezés-feldolgozó egységének"* struktúráját ismertetjük. A kifejezés-feldolgozó egység egy regiszternégyesből felépített hierarchikus struktúra volt, később egy bonyolultabb szerkezetű, de kevesebb regisztert tartalmazó változat került megtervezésre. A regiszternégyesek egy- egy baloperandusz-regiszterből (BOR), operátor-(műveletijel-) regiszterből (OR), jobboperandusz-regiszterből (JOR) és eredményregiszterből (ER) állnak (lásd 3. ábra; az ábra közepe nem jelent regisztert).

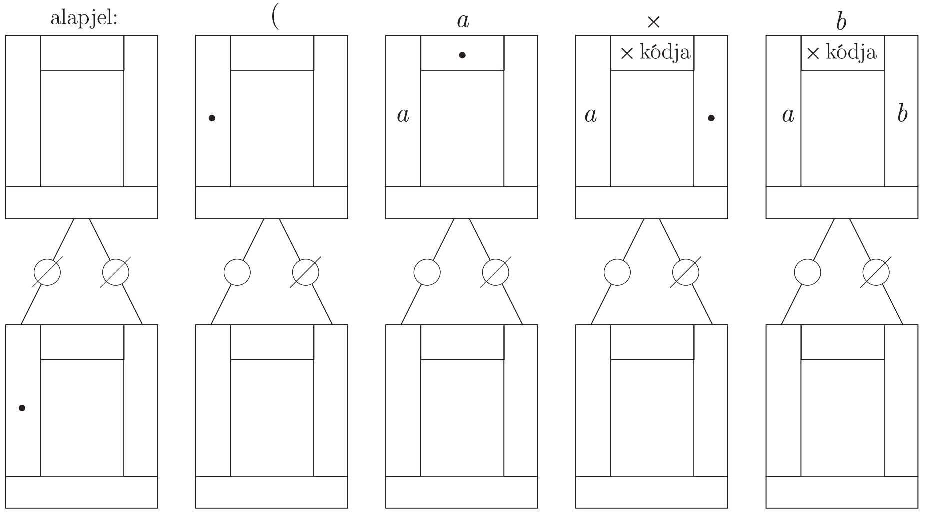


3. ábra

A bal- és jobboperandusz-regiszterek, valamint az eredményregiszterek hossza a gép szóhosszával egyezik meg, az operátorregiszter hossza pedig a gép nyelve alapjeleinek (bináris) numerikus kódja hosszával. A hierarchikus struktúra abban áll, hogy a regiszternégyesek „emeletekbe" vannak rendezve, és a legalsó kivételével minden egyes emeleti ER egy-egy kapuáramkörrel össze van kötve az eggyel alacsonyabb emeleti BOR-ral és JOR-ral úgy, hogy e két kapu közül legfeljebb az egyik lehet nyitva. Az összekötés azt jelenti, hogy a kérdéses ER tartalma automatikusan betöltődik az eggyel alacsonyabb emeleti BOR és JOR közül abba, amellyel nyitott kapuval van összekötve (ha van ilyen), majd a kérdéses ER emeletén mind a négy regiszter kiürül, és az addig nyitott kapu bezárul. A regiszternégyesek ugyanakkor a gép aritmetikai egységének műveletvégző részével is össze vannak kötve. Ezen összeköttetés abban áll, hogy valahányszor valamely JOR megtelik (vagyis az üres állapotból valamely számot tartalmazó állapotba megy át), a kérdéses emeleti BOR és JOR tartalma automatikusan betöltődik azon műveletet végző alegység megfelelő bemenő regisztereibe, amely művelet jelének numerikus kódja megegyezik a kérdéses emeleti OR tartalmával, majd, miután ez az alegység végrehajtotta a kérdéses műveletet, ennek eredménye automatikusan betöltődik a kérdéses emeleti ER-be (ahonnan azután, amennyiben ez az ER nyitott kapuval van összekötve az eggyel alacsonyabb emeleti BOR és JOR valamelyikével, a fentiek szerint abba töltődik be; amennyiben az eggyel alacsonyabban lévő emeleti JOR-ba, akkor ez a folyamat eggyel alacsonyabb emeleten ismétlődik).

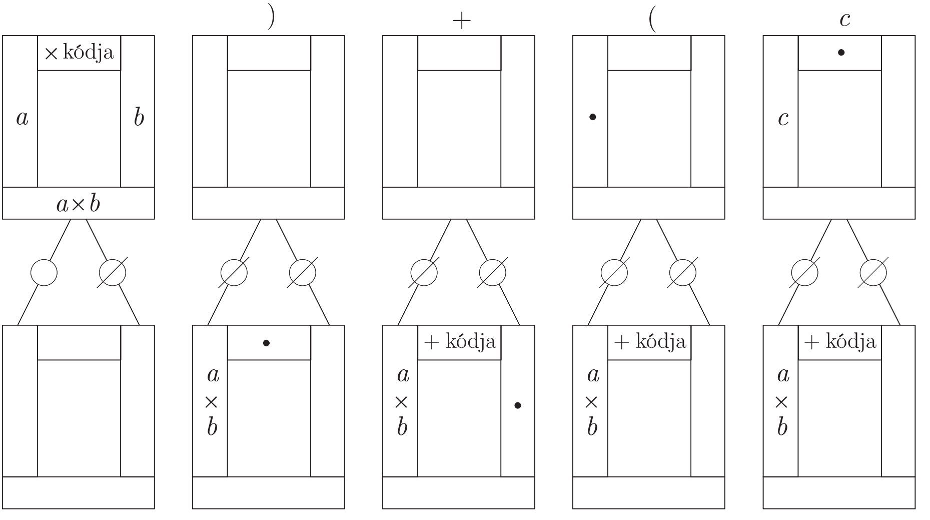
Ebből világos, hogy lényeges valamely regiszter üres (alap-) állapotát valamely számot (akár a 0-t) tartalmazó állapotától megkülönböztetni. Ezért az előjeles számok direkt kódos ábrázolását használjuk; az „aritmetikai” 0 negatív előjellel van ábrázolva (előjelbit 1, a többi bit 0), míg a pozitív előjelű 0 (minden bit 0) az üres állapot jele. A továbbiakat egy példán mutatjuk be.

Legyen a következő utasítás (aritmetikai kifejezés) a gép számára $(a \times b) + (c/d) \Rightarrow u$. A végrehajtást a 4.a, b, c, d ábrák mutatják.

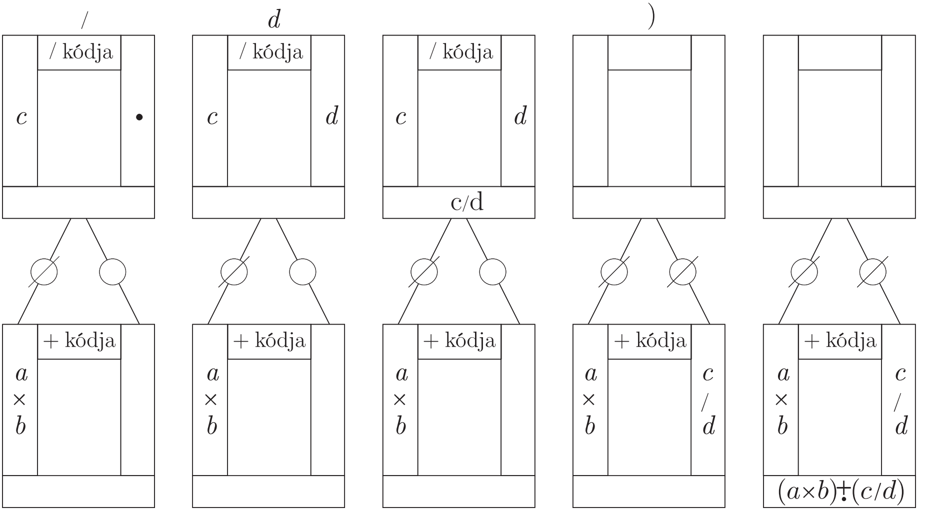


4.a ábra

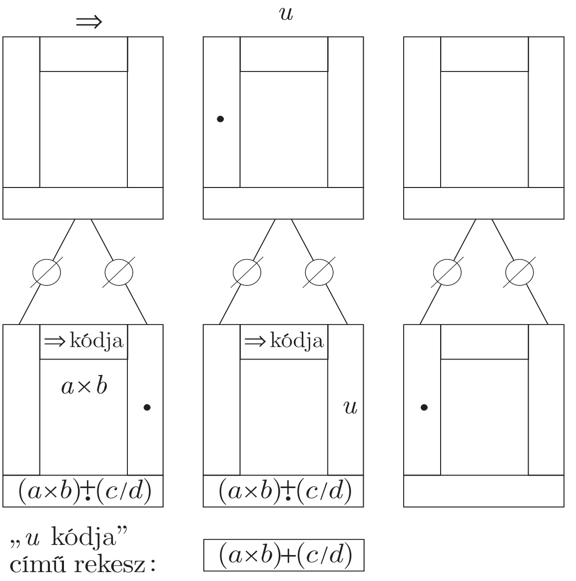
A pont az aktív regisztert jelzi; ○ nyitott, ∅ zárt kaput jelent.



4.b ábra



4.c ábra



4.d ábra

DEFINIÁLT UTASÍTÁSOK I.

ÁLTALÁNOS MEGJEGYZÉSEK

1. UTASÍTÁS ÁLTALÁNOS ALAKJA

INDEXREGISZTER NÉLKÜLI GÉPEN:	INDEXREGISZTERES GÉPEN:
$\nu\Theta \quad \alpha \quad \beta \quad \gamma \quad \ldots$	$\nu\Theta \quad \alpha,\imath \quad \beta\kappa \quad \gamma\lambda \quad \ldots$

ν: változat jele (szám, esetleg vonással, vesszővel stb.)
Θ: művelet jele (egy vagy több betű, az első mindig nagy, egyéb karakterekkel)
$\alpha,\beta,\gamma,\ldots$: címek ($\alpha\beta\gamma\ldots$ címrész; α: első címrész, β: második címrész, ...)
$\imath,\kappa,\lambda,\ldots$: indexek ($\imath$: első index, κ: második index, ...)

2. MEGÁLLAPODÁS KEVESEBB CÍMŰ UTASÍTÁS HASZNÁLATÁRA TÖBBCÍMŰ GÉPEN:

Ha $\nu\Theta$ címnélküli utasítás, akkor $\nu\Theta\ \alpha$, $\nu\Theta\ \alpha\ \beta$, $\nu\Theta\ \alpha\ \beta\ \gamma$ ugyanaz, mint $\nu\Theta$.
Ha $\nu\Theta\ \alpha$ egycímű utasítás, akkor $\nu\Theta\ \alpha\ \beta$ és $\nu\Theta\ \alpha\ \beta\ \gamma$ ugyanaz, mint $\nu\Theta\ \alpha$;
$\tilde{\nu}\Theta\ \alpha\ \beta$ és $\tilde{\nu}\Theta\ \alpha\ \beta\ \gamma$ ugyanaz, mint $\nu\Theta\ \beta$;
$\tilde{\tilde{\nu}}\Theta\ \alpha\ \beta\ \gamma$ ugyanaz, mint $\nu\Theta\ \gamma$.
Ha $\nu\Theta\ \beta$ eredetileg kétcímű utasításként van definiálva, akkor
$\nu\Theta\ \alpha\ \beta\ \gamma$ ugyanaz, mint $\nu\Theta\ \alpha\ \beta$;
$\tilde{\nu}\Theta\ \alpha\ \beta\ \gamma$ ugyanaz, mint $\nu\Theta\ \alpha\ \gamma$;
$\tilde{\tilde{\nu}}\Theta\ \alpha\ \beta\ \gamma$ ugyanaz, mint $\nu\Theta\ \alpha\ \gamma$.

3. REGISZTEREK JELÖLÉSE

E: eredményregiszter (gyűjtős gépen mindig a gyűjtőt jelenti);
A: gyűjtő (akkumulátor, szummátor);
R: szorzó (-osztó) regiszter;
U: utasításszámláló;
Ω: feltételregiszter;
T: túlcsordulás-regiszter;
$|_\imath$: $\imath$-adik indexregiszter.

GÉPI PROGRAMOZÁS I. 1.2.1.01

PROGRAMOZÁSI PÉLDÁK

DIREKT PROGRAMOZÁS I.

1. Feladat.

$$(a+b)(c-d)/(a-b)(c+d) \Rightarrow x; \quad a=(301), b=(302), c=(303), d=(304), x=(305)$$

a) Egycímes gépen (minden egycímű utasítás megengedett):

CÍM	MÜV. JEL	CÍMRÉSZ TART.	CÍMRÉSZ CÍM	MEGJEGYZÉS
201	1B	a	301	a ⇒ (E)
202	1A	b	302	a+b ⇒ (E)
203	1T	m1	351	a+b ⇒ m1 (m1, m2 munkarekeszek tartalma, munkaváltozók)
204	1B	c	303	c ⇒ (E)
205	1S	d	304	c-d ⇒ (E)
206	1MT	m1	351	(a+b)(c-d) ⇒ m1
207	1B	a	301	a ⇒ (E)
210	1S	b	302	a-b ⇒ (E)
211	1T	m2	352	a-b ⇒ m2
212	1B	c	303	c ⇒ (E)
313	1A	d	304	c+d ⇒ (E)
214	1M	m2	352	(a-b)(c+d) ⇒ (E)
215	1'D	m1	351	(a+b)(c-d)/(a-b)(c+d) ⇒ (E)
216	1T	x	305	(a+b)(c-d)/(a-b)(c+d) ⇒ x

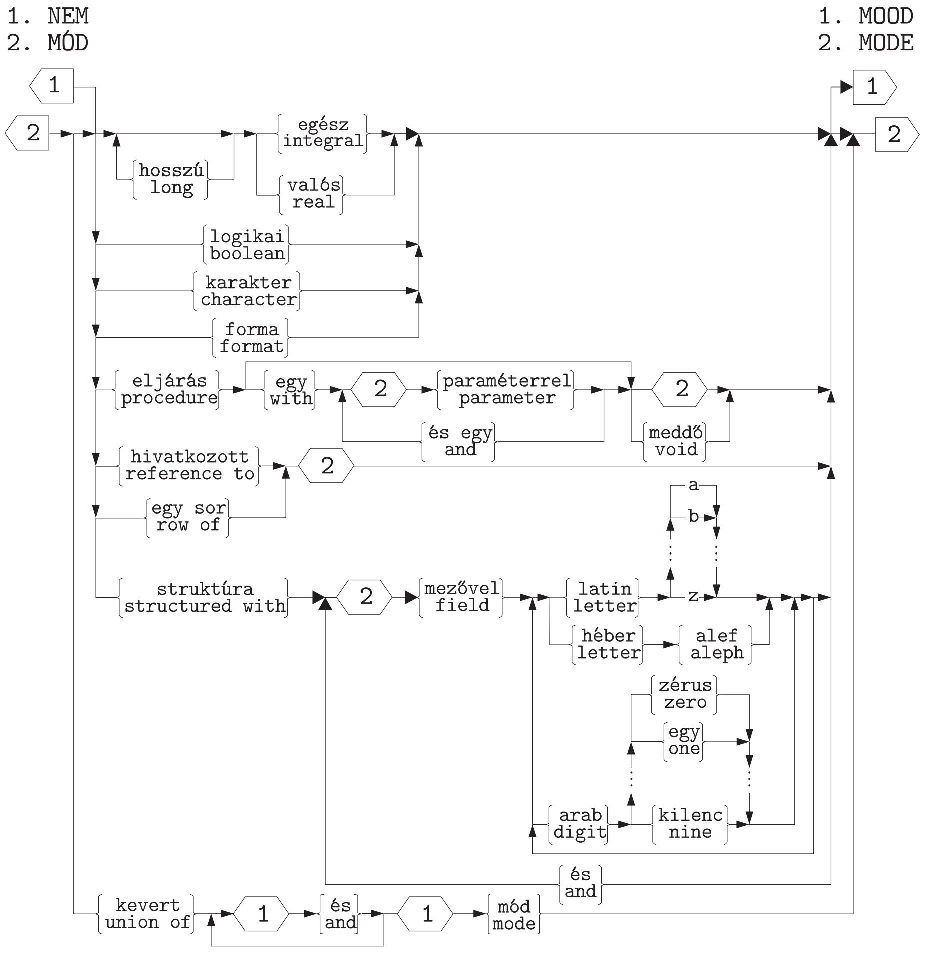
b) Kétcímes gépen (minden egy- és kétcímű utasítás megengedett):

CÍM	MÜV. JEL	I. CÍMRÉSZ TART.	I. CÍMRÉSZ CÍM	II. CÍMRÉSZ TART.	II. CÍMRÉSZ CÍM	MEGJEGYZÉS
201	2A	a	301	b	302	a+b ⇒ (E)
202	1T	m1	351	-	000	a+b ⇒ m1
203	2S	c	303	d	304	c-d ⇒ (E)
204	1MT	m1	351	-	000	(a+b)(c-d) ⇒ m1
205	2S	a	301	b	302	a-b ⇒ (E)
206	1T	m2	352	-	000	a-b ⇒ m2
207	2A	c	303	d	304	c+d ⇒ (E)
210	1M	m2	352	-	000	(a-b)(c+d) ⇒ (E)
211	4'DT	m1	351	x	305	(a+b)(c-d)/(a-b)(c+d) ⇒ x

c) Háromcímes gépen (minden, eredményregiszterre nem hivatkozó egy- és kétcímű és minden háromcímű utasítás megengedett):

CÍM	MÜV. JEL	I. CÍMRÉSZ TART.	CÍM	II. CÍMRÉSZ TART.	CÍM	III. CÍMRÉSZ TART.	CÍM	MEGJEGYZÉS
201	3AT	a	301	b	302	m1	351	a+b $\Rightarrow$ m1
202	3ST	c	303	d	304	m2	352	c-d $\Rightarrow$ m2
203	3MT	m1	351	m2	352	m1	351	(a+b)(c-d) $\Rightarrow$ m1 (lehetne 2MT-vel)
204	3ST	a	301	b	302	m2	352	a-b $\Rightarrow$ m2
205	3AT	c	303	d	304	m3	353	c+d $\Rightarrow$ m3
206	3MT	m2	352	m3	353	m2	352	(a-b)(c+d) $\Rightarrow$ m2 (lehetne 2MT-vel)
207	3DT	m1	351	m2	352	x	305	(a+b)(c-d)/(a-b)(c+d) $\Rightarrow$ x

ALGOL-68

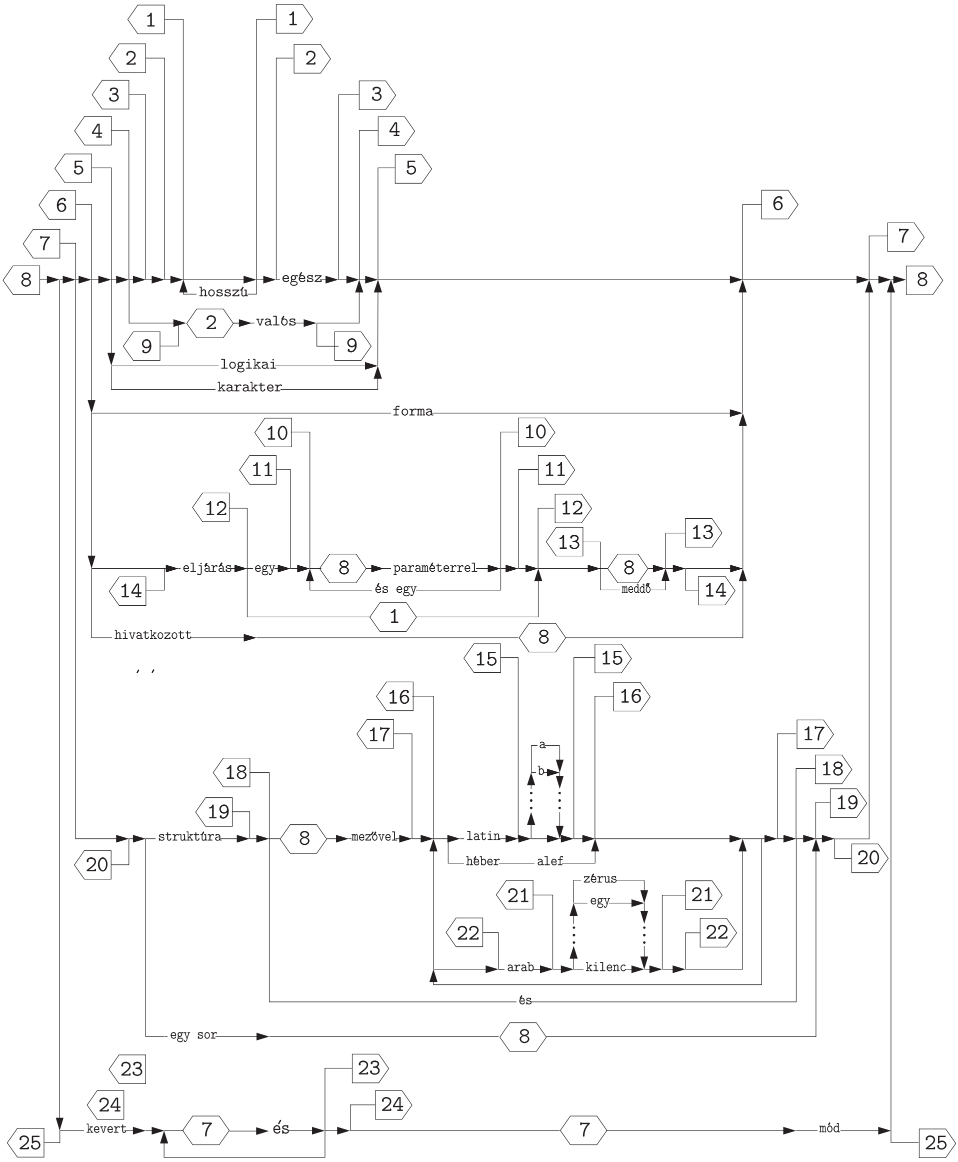


5. ábra
A módok áttekintő zászlós ábrája

ALGOL-68

1. ÜRES	1. EMPTY
2. HOSSZINCS	2. LONGBETY
3. EGÉSZ	3. INTEGRAL
4. ARITMETIKAI	4. INTREAL
5. EGYSZERÜ	5. PLAIN
6. TÍPUS	6. TYPE
7. NEM	7. MOOD
8. MÓD	8. MODE
9. VALÓS	9. REAL
10. PARAMÉTER	10. PARAMETER
11. PARAMÉTEREK	11. PARAMETERS
12. PARAMINCS	12. PARAMETY
13. MED	13. MOID
14. ELJÁRÁS	14. PROCEDURE
15. ALFA	15. ALPHA
16. BETÜ	16. LETTER
17. JEGY	17. TAG
18. MEZÖ	18. FIELD
19. MEZÖK	19. FIELDS
20. ELRENDEZETT	20. STOWED
21. SZÁMNÉV	21. FIGURE
22. SZÁMJEGY	22. DIGIT
23. BALNEM	23. LMOOD
24. BALNEMEK	24. LMOODS
25. KEVERT	25. UNITED

ALGOL-68



6. ábra

A módok metaprodukciós szabályainak zászlós ábrája

IRODALOM

[1] Goldstine H. H., *A számítógép Pascaltól Neumannig*, Műszaki Könyvkiadó, Budapest, 1987.

[2] Kalmár László, Túlságosan absztrakt-e a modern algebra, (*Előadás*), Balatonvilágos, 1955. (jegyzete Péter Rózsa)

[3] Kalmár László, *A matematikai logika műszaki alkalmazásai I.*, (Kézirat), Szeged, 1956.

[4] Kalmár László, *A matematikai logika műszaki alkalmazásai II.*, (*Kézirat*), Szeged, 1956.

[5] Kalmár László, *Az egycímű programozás elemei* (A Szegedi Tudományegyetem Bolyai Intézete Kibernetikai munkacsoportjának keretében tartott programozási tanfolyam anyaga, (*Kézirat*) Szeged, 1956—57.

[6] Kalmár László, A szegedi logikai gép (előadás-kivonat) *Mat. Lapok* **9**(1958), 165.

[7] Kalmár László, An argument against the plausibility of Church's thesis, Constructivity in Mathematics (*Proc. Coll. Amsterdam, 1957*), North-Holland, 1959, 72—80.

[8] Kalmár László, A new principle of construction of logical machines, 2-e Congès Unternat. de Cybernètique (Namur, 1958), 458—463.

[9] Kalmár László, A practical infinitistic computer. Infinitistic Methods in the Foundations of Mathematics (*Proc. Sympos. Warsaw, 1959*), 1961, 347—362.

[10] Kalmár László, On a digital computer which can be programmed in a mathematical formula language, *A II. Magyar Matematikai Kongresszus* (*Budapest, 1960*) előadáskivonatai 5. kötet, 3—16. (Orosz fordításban is megjelent.)

[11] Kalmár László, Un modele algébrique de calculatrice Électronique, Troisième Congrés de Calcul et de Traitement de l'Information (Toulouse, 1963), Paris, 1965, 381—387.

[12] Kalmár László, *Programozás 1962/63* (Előadásjegyzet).

[13] Kalmár László, *Matematikai és nyelvi struktúrák, Általános nyelvészeti tanulmányok, II.* Budapest, 1964, 11—74, 166—172, 295—304.

[14] Kalmár László, Les calculatrices automatiques comme structures algébriques, Prévions, Calcul et Réalités, Paris, 1965, 9—22.

[15] Kalmár László, *Programozás 1966/67*. (Jegyezte Kalmár Lászlóné.)

[16] Kalmár László, Le langage comme structure algébrique. Cahiers Linguist. *Théor. Appl.* **4**(1967), 73—82.

[17] Kalmár László, An intuitive representation of context-free languages. *Internat. Conf. Comput. Linguist. Stockholm, 1969*. Preprint 66. (10 oldal)

[18] Kalmár László, Beszélgetés a matematikáról, *Természet Világa* **103**(1972), 351—356.

[19] Kalmár László, Az elektronikus digitális számítógépek eddigi fejlődése és a várható fejlődés fő irányai. (Kalmár szerkesztésében 1972-ben az MTA Természettudományi I. Főosztálya megbízásából készült tanulmány. Munkatársak: Hunya Péter, Kertész Ádám, Quittner Pál, Sára Attila, Székely Sándor.) 1—70.

[20] Kalmár László, Belső gépi nyelvek, beleértve a magasszintű nyelveket. (Kalmár szerkesztésében 1973-ban a MTA Természettudományi I. Főosztálya megbízásából készült tanulmány. Munkatásak: Gyurkovics Éva, Hunya Péter, Komor Tamás, Makay Árpád, Muszka Dániel, Révész György, Sára Attila, Simon Endre, Székely Sándor, Varga Antal, Varga Tibor.) 1—14.

[21] Kalmár László, A számítástechnikai szakemberképzés problémái. A számítástechnikai oktatás a hazai felsőoktatási intézményekben (Konferencia, Visegrád, 1974), Budapest, 1974. 25—30.

[22] Kalmár László, Géptől független szemlélet kialakítása a programtervezők oktatásában. A számítástechnikai oktatás a hazai felsőoktatási intézményekben. (Konferencia, Visegrád, 1974), 142—146.

[23] Kalmár László, *Integrállevél*, Gondolat Kiadó, Budapest, 1986.

[24] Kemeny, J. G. *Az ember és a számítógép*, Gondolat Kiadó, Budapest, 1978.

[25] Neumann János élete és munkássága, (Neumann János Számítástudományi Társaság), Budapest, 1979.

[26] Péter Rózsa, Kalmár László matematikai munkássága, *Mat. Lapok* **6**(1955), 138—150.

[27] Tarján Rezső, *Kibernetika*, Studium Könyvek 43, Gondolat Kiadó, Budapest, 1964.

[28] Varga Antal, Tizenöt éve halt meg Kalmár László, *POLYGON* **I/1**, Szeged, 1991.

[29] Varga Antal, Neumann János „Hazánk legnagyobb Jancsija" *POLYGON* **IV/1**, Szeged, 1994.

Varga Antal, JATE Bolyai Intézet