

Makay Árpád: Kalmár László számítástechnikai munkássága, (Közlés alatt, TCS), 2004.

Ez a cikk Kalmár László centenáriumi évfordulójára készült, angol nyelvű változata a Theoretical Computer Science (TCS) külön számában jelenik meg. A szerző Kalmár László számítástudományi eredményeit összefoglaló jelleggel ismerteti. A hivatkozott tanulmányok eredetiben a *Tanulmányok* menüpontban lelhetők fel.

A mellékelt, két angol nyelvű Kalmár cikk részletezi az összefoglaló értékelésben szereplő tudományos eredményeket.

Kalmár László, A new principle of construction of logical machines, 2-e Congés Internat. de Cybernetique (Namur, 1958), 458-463.

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.)

Kalmár László számítástechnikai munkássága

Az 1950-es évek végén Kalmár László szerteágazó érdeklődési körébe került a számítástechnika is. Nyilván tudott a számítógépek térhódításáról, a numerikus számításokra való kiváló alkalmazhatóságukról. Mégis valószínűsíthető, hogy a matematika és a számítógépek kapcsolódása az a pont, ami megfogta, és ami később is jellemezte a számítástechnikáról alkotott felfogását. Kétségtelenül matematikusként kezelte a problémákat, a matematika - általa kiválóan ismert - módszereit igyekezett alkalmazni az akkor még tisztán technikai és technológiai területen. Nem sokkal később nyilvánvalóan kiderült, hogy a számítástechnika valóban igényli, és messzemenően hasznosítja a matematika eredményeit gondoljunk csak az automataelmélet vagy a kódoláselmélet inspiratív szerepére. A matematika egzaktága tükröződik a számítástechnikai problémák kezelésében, azok alapos felderítésében, részletes - sokszor aprólékos munkát igénylő - vizsgálatában. Interdiszciplináris tudása alapozta meg a számítástechnika alkalmazhatósági területeinek folyamatos keresését, konkrét feladatok megfogalmazását - amelyekhez sokszor megtalálta a megvalósítás módját és a lelkesen vállalkozó kutatókat, fejlesztőket.

Nem szabad elfelejtenünk, hogy az akkori magyar valóság sokban korlátozta a kutatói, fejlesztői gondolatok szárnyalását. A rendelkezésre álló hazai eszközpark szegényes volt, többségük - talán megérthető okokból - a gazdaság és az államirányítás céljait szolgálta, elenyésző része állt az oktatás és a kutatás rendelkezésére. Kalmár László szerteágazó kapcsolatai - és már az 1960-as években országos elismertsége a számítástechnika területén - segített feloldani a technikai korlátokat, elsősorban az oktatás gyakorlati vonatkozásaiban. Saját elképzeléseire, terveire is rányomta bélyegét az ehhez a szűkösséghez való kényszerű alkalmazkodás. Egyik fő törekvése - a programozási nyelvek közelítése a matematika, ezen belül a logika formalizált nyelvéhez - tulajdonképpen önként vállalt korlátok között haladt. A korlátokat a megvalósítás igénye és a - remélt - rendelkezésre álló erőforrások állították fel.

Semmi nem ingatta meg abban a hitében, hogy a hazai számítástechnika innovatív módon hozzá tud járulni ahhoz a hatalmas fejlődéshez, amit nyomon követett. Az akkor élenjáró folyóiratok, szakkönyvek - nemzetközi ismertsége miatt sokszor különlenyomatokként, tiszteletpéldányokként - rendelkezésre álltak a könyvtárban. Számára majdnem teljesen nyitva volt a világ mindkét fele. A terület élvonalbeli kutatóival a kapcsolatait konferenciákon tudta ápolni, törekvéseit, elképzeléseit ott előadhatta. A perspektivikus kutatási, fejlesztési területekről így közvetlen ismeretei voltak, amit tanítványaival, munkatársaival azonnal megosztott.

Amikor a kezünkbe kerül evv olyan hatékony eszköz, mint a számítógép, sokszor elfut velünk a fantáziánk.

Megtörtént ez a hazai számítástechnika korai szakaszának kutatóival, mérnökeivel, alkalmazóival is - Kalmár Lászlóval is. Közvetlen és azonnal megvalósíthatónak látszott például a gépi fordítás. Az erre alakult csoportot lelkes figyelemmel kísérte és támogatta Kalmár László. Tudjuk, hogy a cél - részben - sokkal később vált elérhetővé. A cél elérhetetlen volt, de a hozzá vezető út gazdagon gyümölcsözött a nyelvtudomány számára.

Kalmár Lászlótól nem álltak távol a filozófiai kérdések. Talán az alapozta meg az ilyen irányú érdeklődését, hogy a században a matematikai logika - teljes matematikai egzaktussággal - is néhány „végső” kérdést feszegetett, többek között a matematika megbízhatóságának határait tapogatta. A számítástechnikában - gyakori szóhasználatban, a kibernetikában számolatlanul merültek fel az olyan kérdések, mint ember és „gondolkodó” gép viszonya, önreprodukció-képesség vagy a számítógép kontrollálhatósága. Kalmár László kialakította és a megfelelő fórumokon el is mondta - véleményét ezekről a kérdésekről. Jellemzően matematikusként, aki messze áll a túlzásoktól, a sci-fi világtól.

Kalmár László a matematikai logika tisztán elméleti kérdéseiben alkotott, a szó legigazibb értelmében alapkutató volt. A számítástechnikában viszont mindjárt kezdetekkor „kézzelfogható” alkotásokra vágyott. Megépíthető - méghozzá az akkori idők technológiájának alapján - eszközöket tervezett.

Az első - megvalósult - terv egy tanszéki szeminárium mellékterméke volt. A téma a matematikai logika - ítéletkalkulus - műszaki alkalmazásai volt. Csábító, ha meggondoljuk, hogy a számítógép aktív részei logikai áramkörök és az általuk végzendő tevékenységek is logikai műveletek. Meg is született az elhatározás ítéletkalkulusbeli formulák értékeit számoló „logikai gép” felépítésére.

A formula legfeljebb 8 változót tartalmazhatott, az ítéletkalkulus szokásos elemi műveleteiből tetszőlegesen összeállítható volt. A változó, ill. egy részformula értékét egy kettős érintkező reprezentálta: egyik pólusa az igaz, másik pólusa a hamis értéknek felelt meg. Egy kétváltozós művelethez ennek megfelelően két bemeneti és egy kimeneti érintkező tartozott. Az érintkezők pólusait huzalok kötötték össze. Az összeköttetéseket a megvalósítandó logikai művelet kívánalmai szerint kellett beállítani, mondhatjuk azt, hogy „programozni”. Az elemi műveletekre „programozott” dobozokból ezután a dobozok kimeneti, ill. bemeneti pólusainak megfelelő összeköttetéseivel - újabb „programozással” - tetszőlegesen hosszú, műveleti prioritásokat figyelembevevő és zárójelezett formula összeállítható. Vegyük észre, a gép nem kerüli meg a problémát valamilyen normálformára redukálással. A legfeljebb 8 változó értékét egyszerű kapcsolóssal állíthatjuk, értékeiket és a hozzájuk tartozó formula-értéket kijelzőkön láthatjuk.

A logikai gép demonstratív eszköz volt, önbizalmat adott a nagyobb feladatok kitűzésére. Minden alkotó megpróbálja megválaszolni azt a kérdést is, hogy alkotása mire jó: a dobozokból összeállított formula egy „áramkör”, amit a gép „tesztel”. Ezt a tevékenységét egy jelfogós „memória” és több formula egyidejű kezelhetősége támogatta. 1959-re egy számítógépekben szükséges alkalmazásra is kibővült a logikai gép: bináris összeadóvá fejlődött.

Nagyjából ettől az időtől kezdve az egyetemen folyamatosan és növekvő hallgatói létszámmal futott a „programtervező matematikus” képzés. Nagyon alapos matematikai ismeretek és programozás - kezdetben assembly szinten, később magasabb szintű nyelveken. Kalmár László matematikai tárgyak mellett a programozás professzora volt.

Az 1960-as években még azt tapasztalhatta, hogy az assembly - még inkább a direkt bináris kódban írt - programok messze hatékonyabbak a fordítóprogramok által produkált kódoknál, nem is szólva a fordítás idő- és memória-igényeiről. A hatékonyság a programozó által a memória-takarékosság és a futási idő optimalizálására fordított munkából eredt. Azt is látta, hogy a programozási munka - a feladat megoldására fordított emberi energia - viszont a magasabb szintű nyelvek használata mellett hatékonyabb. Ma már azt mondanánk, ez utóbbi a fontos. De Kalmár még nem láthatta, hogy nagyjából 25 év múlva a számítógépek memóriája, számítási kapacitása szinte korlátlan lesz, a fordítóprogramok - az idők során született elméleti kutatások eredményeként - intelligenciája ugrásszerűen megnő. A megoldást abban látta, hogy közelíteni kell a számítógépek gépi kódját a magas szintű nyelvek szintaxisához, szemantikájához.

Az ilyen, ún. formulavezérlésű számítógép terveinek kidolgozására munkássága során többször visszatért. A már említett megvalósíthatóságra törekvése okán az egyes verziók tükrözték az aktuális technikai színvonal lehetőségeit. Megfigyelhető az is, hogy ezzel párhuzamosan többször megkísérelte a számítógépet algebrai struktúráként definiálni és terveit erre az egzakt alapra építeni. Például egy műveletet a számítógépben - többek mellett - a memóriatartalom transzformációjának tekintett. A transzformáció tulajdonságait azonban a műveletek nagy száma és bonyolultsága miatt nem lehetett matematikai pontossággal leírni. Ehhez a modell egyszerűsítése kellett volna - például a Turing gép matematikailag megfogható szintjére -, amit a gyakorlat követelményei nem engedtek meg. Így az algebrai modell inkább szemléletet adott, kevésbé a tervezést közvetlenül segítő eszközkészletet.

Az első tervek az ún. Ljapunov-féle operációs nyelvet interpretáló gépre készültek. A nyelv korlátozott számú változót, aritmetikai műveletekből felépülő kifejezéseket és kevés algoritmikus eszközt - feltételes utasítás, ciklus - engedett meg. Szintaxisa egyszerű volt. Lexikális elemzésre szinte nem volt szükség. A blokk-struktúra hiánya miatt a mai interpreterekben elkerülhetetlen - és az akkori technikai színvonalon áramkörökben minden

mate és más interpretációnak emelése - és az akkor technikai szinten nem minden bizonnyal megvalósíthatatlan - nehéz technikák alkalmazását nem igényelte.

A gép végrehajtó egysége regiszternégyesekből veremszerűen felépülő struktúra. A regiszternégyes kétváltozós aritmetikai és logikai műveletek végrehajtására van felkészítve: tárolja a két operandust, a műveletet és - mindkét operandus értékének rendelkezésre állásakor - a kiszámított eredményt. Egy regiszternégyes eredmény regisztere a felsőbb szintű regiszternégyes bal- ill. jobb operandus regisztereivel egy-egy kapun keresztül csatlakozik. A kapuk közül legfeljebb egyik nyitott, ezen fut az alsóbb szintű eredmény a felsőbb szintű operandus regiszterébe.

A program végrehajtása a program jeleinek egymás utáni olvasásával, egy menetben történik. Mindig van egy aktuális regiszternégyes és annak egy aktuális regisztere. A soron következő jel értéke - ha van, azaz a jel változó - bekerül az aktuális regiszterbe. Minden jel esetében újra kijelölésre kerül az aktuális regiszternégyes, annak regisztere és beállítódik a kapuk nyitottsága, ill. zártsága. Ha egy regiszternégyesben elvégezhető a művelet - egyik operandus sem üres -, a művelet végrehajtódik, a nyitott kapun az eredmény felfelé áramlik. Ha utánagondolunk, látjuk, hogy egyszerű ciklusok interpretációját is segíti a fenti regiszter hierarchia, hiszen az ismétlési feltétel - logikai kifejezés - értékének kiszámítására is alkalmas. Ezen felül azonban szükség van vezérlő egységre, amelynek alapvetően a szekvenciális, ill. feltételes és ciklikus végrehajtás interpretáció a feladata. A memóriakezelés, ezen belül az indexes változók kezelése úgyszintén speciális tervezést igényel. Kalmár tervei kitérnek ezekre, elsőrendű eredménye azonban a technikailag kivitelezhető, speciális verem architektúra. A gép hazai megépítésének feltételeit nem sikerült megteremteni, Kalmár gépének egyes elemei azonban beépültek az Ukrán Tudományos Akadémia 1966-ra elkészült MIR gépébe.

Az 1960-as évek elejére már kiderült, hogy a verem szinte mindenható eszköz a számítástechnikában. Ha a hagyományos - infix - kifejezéseket postfix formára konvertálva verembe helyezzük, interpretálásuk gyerekjáték. Maga a konverzió algoritmus is egyszerű, ha eközben vermet használunk. Ha sikerülne az olyan programozási nyelvek, mint pl. az ALGOL-60 algoritmikus elemeit is bekényszeríteni a postfix formába, akkor a program egyszeri transzformációjával - fordítás - akárhányszori végrehajtása a klasszikus gépi kódra való fordítás nélkül, interpretatív módon lehetséges. Többé-kevésbé ez sikerült és ezen az alapon rendkívül hatékony interpreterek születtek. A verem hardver szintű beépítése a számítógépek architektúrájába, a mikroprogramozás technikája tovább növelték a hatékonyságot.

Közben olyan elméleti eredmények születtek, amelyek a számítástechnika fejlődésének irányát egyértelműen megszabták. A véges automaták elmélete megalapozta a lexikális elemzés hatékony algoritmusainak megtervezését. A veremautomaták és speciális osztályaik elmélete megszabta azokat a korlátokat, amelyeken belül kell tartani a programozási nyelvek szintaxisát ahhoz, hogy hatékony elemzők készülhessenek. A korlátok tágak, a programozási nyelvekkel szemben támasztott programozói követelmények bőven kielégíthetőek. Az irány tehát a magasszintű algoritmikus nyelvek - általános célúak és speciális feladatokra alkalmasak egyaránt - használata és hatékony fordítóprogramok, interpreterek használata. 1973-ban Kalmár László a Magyar Tudományos Akadémiától felkérést kapott, hogy tekintse át a számítógépek, a magasabb szintű programozási nyelvek - beleértve a gépi nyelveket is - helyzetét, fejlődését. A cél az volt, hogy a magyar számítástechnika megkeresse ezen a területen a helyét, a világosan jövőbe mutató számítástechnikai kutatásban és fejlesztésben a szerepkörét. Itthon is látszott már, hogy a számítástechnika jelentőségét nem ismertük fel idejében, legalábbis nem fordítottunk rá kellő anyagi erőforrásokat - együtt egész Kelet-Európával. A felkérés nyilván annak szólt, hogy Kalmár azon kevés kutatócsoport egyikének vezetője volt, ahol a kevés támogatás mellett is születtek eredmények, voltak elképzelések.

Az elemző tanulmányosorozat - ma is helytállóan - áttekinti a számítógép-generációk addigi fejlődését, a számítógép-architektúrák egyes elemeinek megvalósítási módjait. Természetesen a Kalmár érdeklődésének középpontjában álló programozási nyelvekre és azok interpretációs lehetőségeire esik a hangsúly. Az utolsó kötetben - a formulavezérlésű gép első tervei után másfél évtizeddel - korszerűsített formában újra javaslatot tesz magas szintű gépi nyelven programozható számítógépre.

Melyek voltak azok az új kihívások, amelyekre az újabb terv reagálni akart?

Áttekintve az algoritmikus - eljárás-orientált - nyelveket, azt látjuk, hogy kikristályosodtak a szintaktikus építőkövek, azokból tulajdonképpen ízlés szerint válogatva tervezhetőek a nyelvek. A blokkstruktúra, a modulokból építkezés feltétlen igény. Szemantikailag a kifejezés fogalma rendkívül kiteljesedett, elsősorban az adatstruktúrák bővülése következményeként. Az alkalmazói igények már messze jártak az 1-2 változós aritmetikai-logikai kifejezésektől. A programok maguk definiáltak adat-típusokat, végrehajtásuk során nem lehetett ismert értékalmazatokra hagyatkozni. Elvált a definíció és az implementáció - amit ma természetesnek talál egy C++ vagy Java programozó. Látszott, hogy komolyan kell foglalkozni a referenciatípussal - ami korábban csak az indexes változónál és az értékadásnál igényelt kezelést. Nem állíthatjuk, hogy mindezekre adekvát feleletet tudott adni az újabb terv, de sokra igen.

A technológia is fejlődött. A verem hardver-kezelése már nem okozott gondot. A mikroprogramozás technikája rendelkezésre állt - mint olyan eszköz, amellyel a hardver architektúra fölé emelhető a programozási szint. Újra reményteliesnek tűnt a megvalósítás.

reményteljesnek tart a megvalósítás.

A tervezéskor Kalmár tisztában volt azzal, hogy adott architektúrán egyetlen algoritmikus programozási nyelv már nem elégíti ki a felhasználókat. Fordítóprogramokra tehát szükség van. A javasolt magas szintű nyelv szerepét így abban határozta meg, hogy álljon közelebb szintaktikai értelemben mindenképpen - az általános célú programozási nyelvekhez. A fordítás így egyszerűbb, a befektetendő munka kevesebb, a hardver-igény alacsonyabb.

A vezetése alatti kutatócsoport részletesen kidolgozta a javasolt nyelvet. Nem tesz konkrét műszaki tervet mellé, mint azt a korábbi formulavezérlésű gépnél tette. Számít arra, hogy a kialakulóban lévő hazai műszaki műhelyek hozzá fogják tenni szakértelmüket, ezekkel együttműködve valósulhatnak meg tervei. A nyelv verifikálására, alkalmasságának bizonyítására szimulációs módszereket javasol. Kutatócsoportjában megvolt az ehhez szükséges szakismeret, de a szimulátor nem készült el. Kalmár energiáit valószínűleg lekötötte elképzeléseinek akadémiai és műszaki körökben való ismertetése. A fogadtatás megfelelő volt, az észrevételek szakmaiak. De éppen ebben az időben Kelet-Európa óriási számítástechnikai fejlesztésekben gondolkodott, Magyarország pedig nem kívánt kimaradni, inkább alkalmazkodott a „nagyok” által meghatározott irányokhoz.

Mit mondjunk ma a „formulavezérlés” gondolatköréről?

Harminc évvel ezelőtt a számítástechnika gyorsan fejlődött, de ilyen iramra senki nem gondolhatott. A múlté a hardver erőforrásokkal való takarékoság, az alkalmazások igényei diktálják a tempót. Sokan helytelenítik ezt a hozzáállást. Nemhogy emelkedett volna a legalacsonyabb programozási szint, még csökkent is, pl. a RISC technológiával. Az alkalmazások hordozhatóságának igénye különböző architektúrák fölött mégis kitermelt egy közös „nyelvű” gépet, a Java végrehajtó gépet. Más célból, más eszközökkel, de a formulavezérlésű géppel közös gyökerekkel.

Szeged, 2004. november 15.

Makay Árpád