Ésik Zoltán

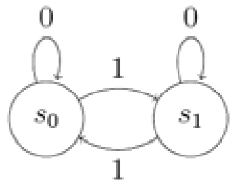
A számítástudomány matematikai alapjai

Ebben a cikkben a teljesség igénye nélkül szeretnénk betekintést nyújtani az 1970-es évek elejétől a Szegedi Tudományegyetemen a számítástudomány matematikai alapjaiban folytatott kutatás eredményeibe.

Az (elméleti) számítástudomány a matematikában, ezen belül a matematikai logikában alakult ki az 1950-es években. Kezdetben olyan neveket említhetünk, mint Stephen C. Kleene, Noam Chomsky vagy Neumann János, de a gyökerek visszanyúlnak az 1930-as évekre, Kurt Gödel, Alonzo Church, Alan Turing munkásságára. Ebben az időszakban fogalmazták meg először matematikai pontossággal azt, hogy mit értsünk algoritmuson, mely problémákat nevezzünk algoritmikusan megoldhatónak. Ez utat nyitott olyan matematikai tételek bizonyításához, amelyek azt mondják ki, hogy valamilyen probléma, feladat, algoritmikusan (és ezért számítógéppel) megoldhatatlan. Ilyen feladat az is, hogy egy adott kijelentés, mint pl. „végtelen sok ikerprím van" (amelyet az elsőrendű logika nyelvén is megfogalmazhatunk), érvényes-e a nemnegatív egész számokra. Itt arról van szó, hogy nem létezik olyan eljárás, amely tetszőleges kijelentésre alkalmazva mindig pontosan válaszolja meg azt a kérdést, hogy az adott kijelentés érvényes-e. Kalmár László munkásságának egy részében ehhez a problémakörhöz csatlakozott, pl. új bizonyítást adott az elsőrendű logika algoritmikus eldönthetetlenségére.

Az automaták és formális nyelvek elmélete az 1950-es, 60-as években alakult ki, Stephen C. Kleene és Noam Chomsky munkásságát követően, és a számítástudomány egyik alappillére. Az automatákat Kleene idegi folyamatok modellezésére vezette be, míg Chomsky az általa bevezetett fogalmakat a természetes nyelvek szintaxisának leírására használta. Az automaták és formális nyelvek elméleti eredményei azóta számos más területen is alkalmazásra kerültek: programozási nyelvek, szerkesztőprogramok, fordítóprogramok, szekvenciális áramkörök, hardver- és szoftverrendszerek tervezése és verifikációja, kommunikációs protokollok, gének elemzése, digitális képfeldolgozás, természetes nyelvek feldolgozása stb.

Nagyon általánosan megfogalmazva az automaták olyan diszkréten működő rendszerek modellezésére szolgálnak, amelyek minden pillanatban véges sok lehetséges belső állapot valamelyikében vannak, és amelyek bemenő jelek hatására állapotot váltanak előre adott átmeneti függvény szerint. Ennek megfelelően az automaták ábrázolhatóak olyan véges irányított gráfokkal, melyek csúcsai az állapotoknak felelnek meg, a lehetséges átmeneteket megadó élek pedig bemenő jelekkel címkézettek. Az ábra egy olyan automatát mutat be, amelynek két állapota van, *s*0 és *s*1, és két bemenő jel hatására válthat állapotot. Ezek a 0 és 1 jelek. Ha az automata az *s*0 állapotban a 0 jelet kapja, akkor új állapota az *s*0 lesz (vagyis marad az *s*0 állapotban), míg az 1 jel hatására az *s*1 állapotba kerül (átmegy az *s*1 állapotba).



1. ábra

Az automataelméleti vizsgálatokat Csákány Béla, Gécseg Ferenc és Peák István indították el Szegeden az algebrai kutatásokhoz kapcsolódóan. Az automaták tekinthetők ugyanis olyan véges algebrai struktúrákként, amelyekben egyváltozós műveletek értelmezettek, és így vizsgálatukra alkalmazható az univerzális algebra eszköztára. Egy másik kapcsolódási pontot adnak az automaták és az algebra közt a (véges) félcsoportok. Minden véges félcsoport tekinthető automataként, másrészt minden automatához hozzárendelhető egy véges „transzformáció félcsoport", mely fontos informácókat ad az automata tulajdonságairól. Ez a kapcsolat megtermékenyítően hatott mindkét terület fejlődésére. A kezdeti szegedi eredmények összefoglalása, pl. az automaták és a félcsoportok kapcsolatáról, megtalálható a [9] könyvben.

Ezt követően az a terület, ahol a legmélyebb eredmények születtek, az automaták kompozíciója és dekompozíciója volt. Itt egyrészt az volt a kérdés, hogy hogyan lehet felépíteni egy összetett automatát egyszerűbb komponensekből, másrészt mely automatákat lehet felépíteni automaták egy adott készletéből. Vagy létezik-e az automatáknak olyan véges készlete, ún. teljes rendszere, amelyből mindegyik automata felépíthető, legalábbis lényegében, azaz bizonyos „viselkedési ekvivalencia" erejéig. A viselkedési ekvivalencia hagyományos definíciója azt veszi figyelembe, hogy az automatában mely nyelvek ismerhetők fel. Tekintsük az ábrán szereplő automatát. Amennyiben az *s*0 állapotból egy irányított úton az *s*1 állapotba jutunk, akkor az útnak megfelelő címkesorozat, bináris szó, páratlan számú 1-est tartalmaz. Az ilyen szavak az *s*0 állapotból az *s*1 állapotba vezetnek. Így ha az *s*0 állapotot választjuk kezdőállapotnak és az *s*1 állapotot az elfogadó állapotnak, az automata azon bináris jelsorozatokat (szavakat) „fogadja el", vagy „ismeri fel", amelyek páratlan számú 1-est tartalmaznak. A kezdőállapot és az elfogadó állapotok különböző megválasztásai mellett az automata 4 lehetséges nyelvet ismerhet fel: az üres nyelvet (nincs elfogadó állapot), az összes bináris szavak nyelvét (mindkét állapot elfogadó), valamint a páros, ill. páratlan számú 1-est tartalmazó szavakból álló nyelveket. Az automaták által felismerhető, vagy röviden „felismerhető nyelvek", robusztus tulajdonságokkal és számos jellemzéssel rendelkeznek. Ezek egyike Kleene klasszikus tétele, mely szerint egy nyelv akkor és csakis akkor felismerhető, ha a véges nyelvekből megkapható a reguláris műveletekkel. Ezek az egyesítés, az összefűzés (konkatenáció) és az iteráció. Például a páros számú 1-et tartalmazó bináris szavak nyelve megkapható (0*10*1)*0* alakban. Itt 0*, a 0 egyetlen egybetűs szót tartalmazó nyelv iteráltja, az összes csak 0-t tartalmazó szavak nyelve. 0*10*1 konkatenácóval áll elő a 0* nyelv és az 1 felhasználásával, és az összes olyan bináris szóból áll, amely pontosan két 1-est tartalmaz, a másodikat a szó végén. A (0*10*1)* nyelv az összes olyan szóból áll, amely páros sok 1-est tartalmaz, és ha a szó nem üres, 1-re végződik. A páratlan számú 1-est tartalmazó bináris szavak nyelve felírható (0*10*1)*0*10* alakban.

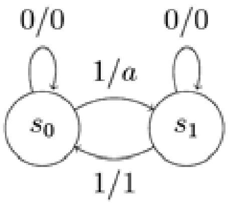
Az automaták kompozíciójának legáltalánosabb formáját Victor Glushkov vezette be. Ez a komponens automaták tetszőleges kapcsolatát megengedi. A.A. Letichevskii egy korai eredménye szerint az általános szorzattal minden automata felépíthető egy olyan automatából, mely irányított gráfja tartalmaz egy olyan erősen összefüggő részt, amely nem egyetlen irányított kör. Az ábra egy ilyen automatát mutat be. A kompozíció egy másik formája az, amikor a komponens automaták úgy függhetnek csak egymástól, hogy nem alakul ki függőségi kör. Ezt hurokmentes kompozíciónak nevezzük. A hurokmentes kompozíció szoros kapcsolatban van a félcsoportok koszorú szorzatával. A hurokmentes kompozícióra vonatkozóan teljes rendszerekről sok fontos információt ad a Krohn–Rhodes-féle felbontási tétel, amely a véges automaták és félcsoportok elméletének egyik híres eredménye. A Krohn–Rhodes-tétel szerint ha egy rendszer teljes a hurokmentes kompozícióra, akkor valamilyen módon minden véges csoport megjelenik a rendszer egy tagjában. Így nem létezik véges teljes rendszer sem. Továbbá jelen van a rendszer valamely tagjában egy bizonyos 3 elemű „egység félcsoport" is. Ugyanakkor megadható automaták olyan rendszere is, amely a Krohn–Rhodes-féle feltételeknek eleget tesz, mégsem teljes. A hurokmentes szorzatra teljes rendszerek leírását a [2] cikkben sikerült megadni.

Automaták hurokmentes kompozíciójában a komponens automaták sorba rendezhetők úgy, hogy minden egyes komponens működését csak a sorrendben megelőző automaták befolyásolják. A Glushkov-féle általános kompozíció és a hurokmentes kompozíció közt vezetett be egy hierarchiát Gécseg Ferenc 1974-ben. Minden egyes i nemnegatív egész számhoz tartozik egyfajta kompozíció, az ún. αi-szorzat. Automaták egy αi-szorzatának komponensei sorba rendezhetőek úgy, hogy minden egyes komponens működését a sorrendben megelőző automatákon kívül az azt követő $i - 1$ automata befolyásolhatja. Amennyiben $i = 0$, akkor a hurokmentes kompozíciót kapjuk. Ha $i = 1$, akkor minden egyes komponens önmagához és a sorrendben megelőző automatákhoz van visszacsatolva. Az $\alpha 1$-szorzat visszavezethető az $\alpha 0$-szorzatra. Ezt felhasználva a teljes rendszerek leírását Ésik Zoltán adta meg. A [4] cikkben pedig bebizonyította, hogy $i \geq 2$ esetén az αi-szorzatra teljes rendszerek éppen azok, amelyek teljesek a Glushkov-féle általános kompozícióra, és amelyeket Letichevszkii már jellemzett. Ez az eredmény felvetette annak lehetőségét, hogy $i \geq 2$ esetén az αi-szorzat ekvivalens az általános kompozícióval. Azt, hogy ez valóban így van, Horváth Gyulával közösen igazolták. Akárhogy is vesszük automatáknak egy készletét, és akárhogyan is rögzítjük az $i \geq 2$ számot, az általános kompozícióval ugyanazokat az automatákat állíthatjuk elő az adott készletből, mint az αi-szorzattal. A bizonyítás egyben egy eldöntési algoritmushoz is elvezetett arra nézve, hogy automaták véges készletéből mely automaták állíthatók elő. Számos kérdés nyitva maradt, ezek közül az egyik legfontosabb az, hogy létezik-e olyan algoritmus, mely automaták egy tetszőleges véges halmazára és egy további automatára eldönti, hogy az előállítható-e $\alpha 0$-szorzattal az adott készletből. (Az $\alpha 1$-szorzatra ez a kérdés algoritmikusan eldönthető.)

Az automaták olyan véges algebrák, amelyekben egyváltozós műveletek értelmezettek. Ezzel szemben a J.E. Doner, valamint J.W. Thatcher és J.B. Wright által bevezetett faautomaták olyan véges algebrák, amelyek tetszőleges (véges) változószámú műveletekkel rendelkeznek. Az elnevezés onnan származik, hogy a faautomaták az univerzális algebrában termeknek nevezett kifejezésekből álló nyelveket ismernek fel, a termeket pedig reprezentálhatjuk címkézett irányított fákkal. Amennyiben csupa egyváltozós műveletünk van, akkor a fának csak törzse van, és így felfoghatjuk szóként is. A faautomaták a fákat kétféle módon dolgozhatják fel, a gyökértől a levelek felé haladva, vagy fordítva, a levelektől a gyökér felé haladva. Mivel a számítástudományban a fákat megfordítva, gyökértől lefelé ábrázoljuk, az előbbit leszálló, az utóbbit felszálló faautomatának nevezzük. Megkülönböztetünk determinisztikus és nemdeterminisztikus faautomatákat is aszerint, hogy adott művelet eredménye egyértékű vagy esetleg többértékű.

Az 1970-es évek második felében Gécseg Ferenc kezdeményezte Szegeden a faautomaták vizsgálatát. Az első Magnus Steinbyvel közös cikkben a determinisztikus leszálló faautomaták minimalizálásával foglalkozott. Nevükhöz fűződik a faautomatákról szóló első monográfia [10] megírása is.

Szegeden a legtöbb eredmény a kimenettel ellátott faautomatákra, az ún. fatranszformátorkra, ill. az általuk megvalósított fát fába képező transzformációkra születtek. A fatranszformátorok a fordítóprogramokban elterjedten használt szintaxisvezérelt fordítási módszer matematikai modelljét alkotják. Automatákra az egyik legalapvetőbb kérdés az ekvivalencia eldöntése. Klasszikus, kimenet nélküli determinisztikus automaták esetében az ekvivalencia eldönthető, sőt, determinisztikus automaták esetében polinomidejű algoritmus is létezik. Kimenettel ellátott automaták esetében összetettebb a kép. Determinisztikus esetben az ekvivalencia eldönthető, míg nemdeterminisztikus esetben algoritmikusan eldönthetetlen (legalábbis abban az esetben amikor egyetlen jel képe egy hosszabb szó is lehet.) Egy kimenettel ellátott automatát mutat be a második ábra. Ha az automata az *s*0 állapotban a 0 bemenő jelet kapja, akkor marad az *s*0 állapotban és a 0 jelet bocsátja ki. Ha az *s*0 állapotban az 1 bemenő jelet kapja, akkor átmegy az *s*1 állapotba, és az *a* jelet bocsátja ki. Az *s*0 állapotból indulva egy bináris szót úgy alakít át, hogy az 1 jel minden páratlan sorszámú helyébe az *a* jelet írja.



2. ábra

Fatranszformátorokra az eldönthetőségi kérdések vizsgálatát Ésik Zoltán kezdeményezte 1978-ban. Egyik eredménye szerint algoritmikusan eldönthető a determinisztikus leszálló fatranszformátorok ekvivalenciája, és az is, hogy egy nemdeterminisztikus leszálló fatranszformátor által indukált transzformáció egyértékű-e, azaz minden bemenő fához csak egy kimenő fát rendel-e. A determinisztikus felszálló fatranszformátorok ekvivalenciájának problémája visszavezethető a determinisztikus leszálló fatranszformátorok ekvivalencia problémájára, az eldönthetőséget ebben az esetben közvetlenül Zachar Zoltán igazolta.

A fatranszformátorokra és fatranszformációkra vonatkozó szegedi eredmények másik nagy része a fatranszformációk kompozíciójával és dekompozíciójával foglalkozik. Joost Engelfriet az 1970-es évek végén bebizonyította, hogy a fatranszformátorokkal indukálható transzformációk nem zártak a kompozícióra, és a kompozíció iterálásával végtelen hierarchiák nyerhetők. Engelfriet nyomán Fülöp Zoltán és Vágvölgyi Sándor vizsgálták a fatranszformátorok és ezek egyes részosztályai által indukálható fatranszformáció-osztályok kompozícióra való lezárását, és több esetben az így nyerhető osztályok teljes leírását adták meg. Eredményeik összefoglalása megtalálható a [8] monográfiában. Vágvölgyi Sándor, részben Fülöp Zoltánnal több eredményt ért el a termátítró rendszerek vizsgálatában is, amelyek a fatranszformátorok általánosításaként is tekinthetők. A termátítró rendszerek a formális algebrai számításokat modellezik, és fontos szerepet játszanak a szimbolikus számítási programcsomagokban és a funkcionális programozási nyelvek értelmező (interpreter) programjaiban. Fülöp Zoltán újabb eredményeiben megjelennek a súlyozott faautomaták és fatranszformátorok is. Egy közönséges fatranszformátor egy bemenő fához kimenő fákat rendel, determinisztikus esetben ez egyértelműen meghatározott. Amennyiben a bemenő fa egy nyelv egy mondatát határozza meg (annak szintaxis fája), akkor a kimenő fák a lehetséges fordításokat adják meg valamilyen nyelven. De sokszor nem elegendő csak a lehetséges fordítások meghatározása, hanem azok közt sorrendet kell meghatározni, vagy a legvalószínűbb fordítást kell kiválasztani. Az ilyen és hasonló kérdések megválaszolására vezették be a súlyozott (fa)automatákat és fatranszformátorokat, melyekre számos eredmény született Szegeden. A súlyozott faautomaták minden bemenő fát valamilyen súllyal fogadnak el, a súlyozott fatranszformátorok pedig minden lehetséges fordításhoz egy súlyt rendelnek. A súlyok egy absztrakt algebrai struktúrát, ún. félgyűrűt alkotnak. A terület eredményeit összefoglaló [3] kézikönyvhöz Ésik Zoltán két, Fülöp Zoltán pedig egy fejezettel járult hozzá. A félgyűrű feletti automaták központi helyet foglalnak el a [6] monográfiában is.

Kleene említett nevezetes tétele a felismerhető nyelvek algebrai jellemzését adja, a felismerhető nyelveket úgy azonosítja, mint a reguláris műveletekkel nyerhető ún. reguláris nyelveket. A felismerhető nyelvek számos más jellemzése is ismert, köztük egy logikai jellemzés, amely J. Richard Büchi, Calvin C. Elgot és Boris Trakhtenbrot nevéhez fűződik. Egymástól függetlenül igazolták azt, hogy szavak felett a monadikus másodrendű logika (az elsőrendű logika monadikus relációváltozókkal való kiterjesztése) ekvivalens az automatákkal: egy nyelvet alkotó szavak tulajdonságai pontosan akkor fogalmazhatóak meg monadikus másodrendű logikában, ha a nyelv felismerhető. Egyben egy eljárást is adtak arra, hogyan lehet automatákat logikai formulákká, ill. logikai formulákat automatává alakítani. (Módszerük katonai célt is szolgált abban az időben a szekvenciális áramkörök tervezésében és gyártásában.)

Mely nyelvek definiálhatóak a monadikus másodrendű logika frágmenseiben? Az elsőrendű logika esetében a választ McNaughton és Papert adták meg: azok a nyelvek, amelyek felismerhetőek olyan (determinisztikus) automatával, amely előállítható hurokmentes szorzattal az egység félcsoportnak megfelelő automatából. A Krohn–Rhodes-tétel szerint ezek ugyanazok az automaták, amelyekben egyetlen bemenő szóra sem képezhető nemtriviális kör, vagy másképp kifejezve, az automata nem tartalmaz nemtriviális csoportot. A monadikus másodrendű logika további speciális eseteit kapjuk akkor, ha az elsőrendű egzisztenciális és univerzális kvantorok helyett reguláris Lindström kvantorokat tekintünk. A McNaughton–Papert-féle jellemzés általánosításaként beláttuk, hogy kölcsönösen egyértelmű kapcsolat létesíthető a monadikus másodrendű logika reguláris Lindström-frágmensei és az automaták bizonyos olyan osztályai közt, amelyek zártak a hurokmentes szorzatra, ld. [7]. Hasonló algebrai jellemzést adtunk a hardver- és szoftverrendszerek verifikálásában használt lineáris temporális logikák kifejező erejére. A vizsgálatokat a 2000-es években Ésik Zoltán és Iván Szabolcs kiterjesztették faautomatákra, fanyelvekre és elágazó temporális logikákra.

Egy függvény fixpontja egy olyan pont, amelyhez a függvény önmagát rendeli. A fixpontok fontos szerepet játszanak a matematikában és a számítástudományban is, az utóbbiban elsősorban azért, mert a rekurzív definíciók szemantikáját függvények, funktorok vagy egyéb konstruktorok fixpontjaiként adjuk meg. A számítástudományban leggyakrabban használt fixpont tételek Kleene és Tarski bizonyos részben rendezett halmazokon értelmezett monoton, ill. rendezésfolytonos függvényekre vonatkozó fixponttételei, melyek szerint ilyen függvényeknek létezik legkisebb fixpontja; a teljes metrikus terek feletti valódi kontrakciókra vonatkozó Banach-féle fixpont tétel, mely szerint ezen függvényeknek létezik egy és csakis egy fixpontja; ill. ezen tételek kategóriaelméleti általánosításai.

Említsünk néhány példát a fixpontok számítástudományi felhasználására. Minden automatához és a kezdőállapot és a végállapotok tetszőleges megválasztásához megadható olyan függvény, amely nyelvekből álló vektorokat képez le nyelvvektorokba, úgy, hogy az automata által felismert nyelv a tartalmazására nézve legkisebb fixpont valamely komponense. Hasonlóan, minden környezetfüggetlen nyelvtanhoz megadható olyan függvény, hogy a nyelvtan által meghatározott nyelv a függvény legkisebb fixpontjának valamelyik komponese. Rendezzük egy halmaz feletti parciális függvényeket gráfjaik tartalmazási relációja szerint, és terjesszük ki ezt a rendezést parciális függvények vektoraira komponensenként. Ekkor minden, valamely funkcionális programozási nyelven megírt rekurzív programhoz megadható olyan funkcionál, azaz parciális függvényeken értelmezett függvény, hogy a program szemantikája, vagyis a program által leírt függvény a funkcionál legkisebb fixpontjának valamelyik komponense. Egy további példával élve, a programozási nyelvekben kiterjedten használt rekurzív adattípusok, mint pl. fák, listák, megkaphatók úgy, mint a halmazok és függvények kategóriája bizonyos funktorainak iniciális fixpontjai. Egy adott abc feletti végtelen szavak a szokásos első eltérésen alapuló metrikával teljes metrikus teret alkotnak, így valódi kontrakciók fixpontjaival definiálhatunk végtelen szavakat, mint pl. a Thue–Morse szót:

$$0110100110010110100101100110100 1\ldots$$

(Az Olvasóban bizonyára felvetődik a képzési szabály megfejtése.)

Fixpontokra és fixpontműveletekre talán az egyik legfontosabb eredmény annak igazolása volt, hogy számítástudományban leggyakrabban használt fixpontműveletek ugyanazoknak az azonosságoknak tesznek eleget. Ezen azonosságok modelljei az iterációs elméletek, melyeket egymástól függetlenül Stephen L. Bloom, Calvin C. Elgot és Jessy B. Wright, valamint Ésik Zoltán vezettek be 1980-ban. (Első cikkében Ésik Zoltán az általánosított iteratív elmélet elnevezést használta.) Bár nem létezik véges azonosságbázis, sikerült az azonosságok teljes leírását adni végtelen sok azonosság segítségével. Ezzel szemben az azonosságelmélet polinomidőben eldönthető, hatékony algoritmust adtunk annak eldöntésére, hogy egy azonosság teljesül-e. Az 1990-es évek elejéig elért eredményeket foglalja össze az [1] monográfia. Azt követően is számos eredmény született, az iterációs elméletek axiómáit is sikerült jelentősen egyszerűsíteni [5], bár valószínűleg további egyszerűsítés is lehetséges. Azt is sikerült megmutatni, hogy ún. kvázi-azonosságok segítségével az iterációs elméletek azonosságai végesen leírhatóak.

Az iterációs elméletek széles körűen felhasználásra kerültek axiomatikus kérdések megoldására. Az eredmények kiterjednek a matematika és a számítástudomány több ágára, pl. az automatákra és formális nyelvekre, relációalgebrákra, részben rendezett halmazokra és kategóriákra, processzus algebrákra, programozási logikákra stb.

Csak néhány eredményt említve, beláttuk, hogy a reguláris nyelvek azonosságai, a binér relációk Kleene algebráinak azonosságai a relációs inverz képzéssel vagy anélkül, a racionális hatványsorok azonosságai, a felismerhető (reguláris) fanyelvek azonosságai, a processzusok biszimulációs és szimulációs ekvivalenciaosztályainak azonosságai, mind relatíven végesen axiomatizálhatóak az iterációs elméletek azonosságainak felhasználásával. Sikerült lényeges haladást elérni az automaták és formális nyelvek és a processzus algebrák axiomatikus alapokra való helyezésében. Pl. Kleene klasszikus tétele és általánosításai, a környezetfüggetlen nyelvekre vonatkozó nevezetes normálforma tételek (Chomsky, Greibach), vagy Rohit Parikh híres tétele, mely szerint minden környezetfüggetlen nyelv betűekvivalens egy regulárissal, mind levezethetők az iterációs elméletek azonosságaiból. Sikerült megmutatni, hogy a programok helyességének bizonyítására szolgáló Hoare-kalkulus helyessége és ún. expresszív struktúrákon való teljessége is származtatható az iterációs elméletek azonosságaiból. A legutóbbi években sikerült általánosítani Tarski és Kleene fixpont tételeit is, és megmutatni azt, hogy a logikai programok szemantikája (negáció esetén is) jellemezhető az új fixpontművelettel.

Néhány könnyen érthető eredménnyel zárjuk a Szegedi Tudományegyetemen elért, a számítástudomány alapjait illető matematikai kutatások eredményeinek bemutatását. Bár ezek először meglehetősen speciálisnak tűnnek, mégis sokkal tovább mutatnak, mivel hozzájárulnak egy új aktív terület, a „hatékony modellelmélet" kialakulásához. Látszólag nagyon egyszerű dologról van szó. Tekintsük egy (véges abc feletti) nyelv szavainak lexikografikus rendezését. Milyen rendtípusok lépnek fel?

Pl. a 1*0 nyelv lexikografikus rendezése

$$0 < 10 < 110 < \ldots,$$

megegyezően a nemnegatív egész számok rendezésével. Tekintsük a 0*1 nyelvet. Ennek rendezése

$$1 > 01 > 001 \ldots,$$

a negatív egészek szokásos rendezése. Végül tekintsük a (00+11)*01 reguláris nyelvet, melyben minden szó u01 alakú, ahol u felbontható x1…xn alakban úgy, hogy minden egyes xi 00 vagy 11. Könnyű megmutatni, hogy a nyelv lexikografikus rendezésében nincs első vagy utolsó szó, és bármely két szó között van egy harmadik. Így a rendezés ugyanaz, mint a racionális számok rendezése. Sikerült jellemezni a reguláris és környezetfüggetlen nyelvek lexikografikus rendezéseit, és megmutatni azt, hogy minden reguláris vagy determinisztikus környezetfüggetlen nyelv elsőrendű, ill. monadikus másodrendű elmélete algoritmikusan eldönthető. Speciálisan, a nemnegatív egészek, vagy a racionális számok rendezésének elmélete eldönthető – ezek Büchi és Rabin híres eredményei. Ugyanakkor van olyan környezetfüggetlen nyelv, amely lexikografikus rendezésének elsőrendű elmélete algoritmikusan eldönthetetlen.

Az 1970-es évektől az eredmények mintegy 400 referált cikkben és 6 könyvben kerültek ismertetésre. A kutatást több mint egy tucat OTKA-pályázat, egy ESF-pályázat és számos bilaterális pályázat (USA, Japán, Ausztria, Németország, Franciaország, Görögország, Finnország) támogatta. Gécseg Ferencet a Magyar és Finn Tudományos Akadémia, Ésik Zoltánt az Európai Akadémia tagjává választották.

Hivatkozások

1. S.L. Bloom, Z. Ésik: Iteration theories, The equational logic of iterative processes, EATCS Monograph Series in Theoretical Computer Science, Springer, 1993.
2. P. Dömösi, Z. Ésik: Critical classes for the α0-product, *Theoretical Computer Science* 61(1988), 17–24.
3. M. Droste, W. Kuich, H. Vogler (szerk.): Handbook of weighted automata, Monographs in Theoretical Computer Science, an EATCS series, Springer, 2009.
4. Z. Ésik: Homomorphically complete classes of automata with respect to the α0-product, *Acta Sci. Math.* (Szeged), 48(1985), 135–141.
5. Z. Ésik: Group axioms for iteration, *Information and Comutation*, 148(1999), 131–180.
6. Z. Ésik, W. Kuich: Modern automata theory, available online, http://www.dmg.tuwien.ac.at/kuich/
7. Z. Ésik, K.G. Larsen: Regular languages definable by Lindström quantifiers, *Theoretical Informatics and Applications*, 37 (2003), 179–241.
8. Z. Fülöp, H. Vogler: Syntax-directed semantics – Formal models based on tree transducers, Monographs in Theoretical Computer Science, an EATCS series, Springer, 1998.
9. F. Gécseg, I. Peák: Algebraic theory of automata, Akadémiai Kiadó, 1972.
10. F. Gécseg and M. Steinby: Tree automata, Akadémiai Kiadó, 1986.