

dc_640_12

Szimbólumfeldolgozó rendszerek leírási
bonyolultsága - Klasszikus és
nem-klasszikus számítási modellek

(The Descriptive Complexity of Rewriting
Systems - Some Classical and Non-Classical
Models)

Vaszi György

doktori értekezésének tézisei

2013.

dc_640_12

1. Bevezetés

A disszertációban klasszikus és nem-klasszikus szimbólumfeldolgozó rendszerek (mint formális nyelveket leíró eszközök, számítási modellek) vizsgálatával foglalkozunk leírási bonyolultságuk, azaz az általuk nyújtott nyelv-reprezentáció tömörségének szempontjából.

Először a formális nyelvek klasszikus elméletéből jól ismert formális grammatikák külső kontroll-mechanizmussal ellátott változatait vizsgáljuk, mely kontroll-mechanizmusok a produkciós szabályok alkalmazhatóságára fogalmaznak meg valamilyen formában további feltételeket, azaz tulajdonképpen a mondatformák levezetése közben történő szabályalkalmazást vezérlik. Ez a fajta vezérlés azért lehet hasznos, mert így egyszerű szabályok (általában környezetfüggetlen szabályok) segítségével lehet leírni bonyolultnak tekinthető nyelveket. A szabályok egyszerűsége egyrészt jobban áttekinthető, jobban kezelhető levezetési folyamatot eredményez, másrészt a vezérlés módosításával, „finom-hangolásával” lehetőség van arra is, hogy csak a kellő mennyiségű „környezetfüggőséget,” a kellő mennyiségű plusz „erőt” adjuk a rendszerhez. További részletekért lásd az [5] monográfiát és a [6] kézikönyv-fejezetet.

A dolgozat következő részében az ún. párhuzamos, kommunikáló grammatikarendszerek tulajdonságaival foglalkozunk. Míg a formális nyelvek hagyományos felfogása szerint egy nyelvet egyetlen grammatikával állítunk elő, addig a grammatikarendszerek elmélete olyan eszközökkel foglalkozik, amelyekben több grammatika egymással kooperálva végzi ugyanezt a feladatot. A párhuzamos, kommunikáló grammatikarendszerek (PC grammatikarendszerek) az osztott szimbólumfeldolgozás formális nyelvi modelljei, működésük során egyszerre több komponens grammatika egymással kommunikálva generálja saját mondatformáját. A kommunikáció speciális nemterminálisok, ún. kommunikációs szimbólumok segítségével történik, a levezetések során az egyes komponensek által generált szavakban megjelenő kommunikációs szimbólumokat helyettesíteni kell a szimbólumhoz tartozó grammatika éppen aktuális mondatformájával. A rendszer által generált nyelv egy előre rögzített komponensnél, a főnöknél megjelenő terminális szavakból áll. A grammatikarendszerek elméletével kapcsolatos részletekért lásd a [2] monográfiát, és a [7] kézikönyv-fejezetet.

A disszertáció harmadik részének témája az úgynevezett nem-hagyományos számítási modellek egyik típusának, a membrán rendszerek (P rendszerek) bizonyos változatainak vizsgálata. A membrán rendszerek fő összetevője egy membrán struktúra, mely hierarchikusan egymásba ágyazott membrán-

nokból áll. A membránok által körbezárt régiókhoz különböző szabályrendszerek tartoznak, melyek a régió tartalmának viselkedését határozzák meg. A régiók objektumok (szimbólumok) multihalmazait tartalmazzák, a régióhoz tartozó szabályok megváltoztathatják az egyes objektumokat (ezek az átírási szabályok), vagy szomszédos régiók között mozgathatják az egyes objektumokat (kommunikációs szabályok). A szabályok alkalmazása, azaz a rendszer működése párhuzamosan zajlik, a tartalmazott objektum multihalmazok evolúciója, működés közbeni változása, a létrejövő konfigurációsorozat egy számítási folyamatnak felel meg. A membrán rendszereknek számos változata ismert, részletekért lásd a [21] monográfiát és a [19] kézikönyvet.

A disszertációban mindhárom fent vázolt téma során egységes célt követünk: azt vizsgáljuk, hogyan lehet korlátozni illetve csökkenteni az egyes modellek méretét befolyásoló paraméterek értékét, azaz az egyes modellek leírási bonyolultságát, illetve milyen viszonyban állnak egymással a különféle, méret-bonyolultságot leíró mértékek, mennyiben befolyásolja az egyik mérték változtatása a többi mérték szerint mérhető komplexitást.

2. A dolgozat célkitűzései

A bevezetésben vázolt számítási modellek vizsgálatával kapcsolatban az alábbi konkrét célkitűzéseket fogalmazzuk meg.

A külső kontroll-mechanizmussal vezérelt környezetfüggetlen grammatikák három különböző változatát vizsgáljuk: a fa vezérelt grammatikákat (tree controlled grammars) [4], az egyszerűsített szövegfeltételekkel vezérelt grammatikákat (simple semi-conditional grammars) [17], és a szétszórt szövegfeltételekkel adott grammatikákat (scattered context grammars) [11]. Az ezen grammatikákkal kapcsolatos leírási bonyolultsági mértékek közül vizsgáljuk a nemterminális szimbólumok számát és a produkciós szabályok számát, illetve a kontroll-mechanizmus bonyolultságának mértékeit: a kontroll nyelv bonyolultságát és a vezérlő szövegfeltételek számát és méretét.

- Hogyan finomíthatók a kontroll-mechanizmusokkal ellátott grammatikák leírási bonyolultságával kapcsolatos eredmények, milyen új technikákkal illetve az ismert technikák milyen módosításával lehet a grammatikák leírási bonyolultságát tovább csökkenteni?
- Mekkora a szétszórt szövegfeltételekkel adott grammatikákban minimálisan szükséges nemterminálisok száma?

A PC grammatikarendszerekben a megkülönböztetjük a kommunikáció két alapvető formáját. A kommunikáció mindkét esetben az egyes grammatikákra mutató kommunikációs szimbólumoknak a kérdezett grammatika aktuális mondatformájával történő helyettesítésével zajlik, de ún. visszatérő rendszerekben kommunikáció után a megkérdezett komponens új levezetésbe kezd, míg a nem visszatérő rendszerekben folytatja megkezdett munkáját.

A visszatérő rendszerekkel kapcsolatban ismert, hogy öt környezetfüggetlen komponens elegendő minden rekurzívan felsorolható nyelv generálásához [3], illetve hogy minden nem visszatérő rendszer szimulálható visszatérő módon úgy, hogy a komponensek száma maximum négyzetesen nő [8, 23]. Az előzőekkel nagyjából egy időben kiderült az is, hogy nem visszatérő rendszerekkel is generálható minden rekurzívan felsorolható nyelv [14], de az ezt bizonyító konstrukció nem mutatott korlátot a szükséges komponensek számára. Később [24] legfeljebb nyolcban határozta meg a szükséges komponensek számát.

- Lehetséges-e nem visszatérő rendszerek komponensszámának további csökkentése, illetve igaz-e hogy a nem visszatérő rendszereknek több komponensre van szükségük mint a visszatérő rendszereknek ahhoz, hogy hasonlóan bonyolult nyelveket generáljanak?

Ha úgy tekintünk a PC grammatikára mint egymással együttműködő ágensek, komponensek közösségére, természetesen adódik az a kérdés is, hogyan szervezhető meg hatékonyabban a köztük zajló kommunikáció.

- Hogyan lehet a komponensek hálózatának kommunikációs struktúráját egyszerűsíteni, egyes komponensek „látókörét” csupán a hozzá valamilyen szempontból közeli, vagy hozzá hasonló komponensekre korlátozni?
- Lehet-e a PC grammatikarendszerek működését úgy szervezni, hogy a rendszer kevésbé határozott, kevésbé célirányos kérdések megfogalmazása mellett is működőképes maradjon?

A membrán rendszerek elméletén belül a disszertációban az ún. symport/antiport rendszereket és a P kolóniákat vizsgáljuk. A symport/antiport P rendszerek olyan membrán rendszerek, melyekben az objektumok megváltoztatása (átírása) nem megengedett, a rendszer konfigurációja csak az objektumok régiók közti mozgásával, azaz kommunikációs szabályok (ún.

symport/antiport szabályok) alkalmazásával változtatható meg [20]. A symport/antiport rendszerek leírási bonyolultságát jellemző paraméterek közül a membránok (régiók) számát és a szabályok méretét, azaz az egy-egy szabály által „megmozgatott” multihalmazok számosságát vizsgáljuk.

- Minimális méretű symport/antiport szabályok használata esetén mekkora méretű rendszer, vagyis hány membrán szükséges ahhoz, hogy a számítási erő ne csökkenjen?

A P kolóniák egyszerű membránok által körbezárt régiókból (cellákból) állnak melyek egy közösen „lakott” környezet részeként egymással és a környezetben megjelenő objektumokkal kölcsönhatásban működnek [12]. A cellák belsejében és a környezetben lévő objektum multihalmazok feldolgozásával létrejövő konfigurációsorozat adja a P kolónia által végzett számítási folyamat jellemzését. A P kolóniák fontos leírási bonyolultsági paraméterei a rendszert alkotó cellák száma, az egyes cellákhoz rendelt különböző evolúciós szabályok, az ún. programok száma és bonyolultsága, valamint az egyes cellák belsejében egyszerre tartózkodni képes objektumok mennyisége.

A P kolóniákkal kapcsolatban a következő kérdéseket vizsgáljuk.

- Hogyan lehetséges a P kolóniák leírási bonyolultsági mértékeit párhuzamosan, azaz mind a cellák mind a szabályok számát együttesen csökkenteni úgy, hogy a rendszer számítási ereje ne csökkenjen?
- Milyenek lehetnek azok a minimális bonyolultságú cellák illetve programok, amelyek használata esetén a P kolóniák számítási ereje még megmarad?

3. Az elvégzett vizsgálatok, elért eredmények

A külső kontroll-mechanizmusokkal vezérelt grammatikákkal kapcsolatban a következő eredményeket értük el. A fa vezérelt grammatikákban szükséges nemterminálisok számát egy korábban ismert technika [22] finomításával sikerült lejjebb szorítani és így a jelenleg ismert legjobb korlátot felállítani.

1. A fa vezérelt grammatikákban minimálisan szükséges nemterminálisok száma nem több mint hat, ebből öt a grammatika produkciós szabályaihoz, egy pedig a kontroll nyelv generálásához szükséges (3.2.1 tétel).

Az egyszerűsített szövegfeltételekkel vezérelt grammatikák produkciós szabályainak leírási bonyolultsága egy (i, j) számpárral jellemezhető, ahol i a megengedő, j a tiltó szövegfeltétel hossza, (i, j) -t a szabály fokának nevezük. Ismert volt [18], hogy 12 legfeljebb $(2, 1)$ fokú feltételes szabállyal minden rekurzívan felsorolható nyelv generálható.

2. Az egyszerűsített szövegfeltételekkel vezérelt grammatikák leírási bonyolultságával kapcsolatban is sikerült a korábban ismert eredményeknél jobb korlátokat felállítani:
 - (a) Ha az egyszerűsített szövegfeltételekkel vezérelt grammatikák feltételes szabályainak foka legfeljebb $(2, 1)$, akkor a feltételes szabályok szükséges száma legfeljebb tíz (3.3.1 tétel).
 - (b) Ha a szövegfeltételek mérete nagyobb lehet, azaz a szabályok foka legfeljebb $(3, 1)$, akkor nyolc feltételes szabály is elegendő a grammatikák maximális generatív erejének eléréséhez (3.3.2 tétel).

Az utóbbi eredmény a disszertáció benyújtásának időpontjában is a legjobb ismert korlátot jelenti a feltételes szabályok számára.

A szétszórt szövegfeltételekkel adott grammatikák leírási bonyolultságát leíró mértékek közül az szétszórt szabályok és a nemterminális szimbólumok számát vizsgáltuk. Korábban ismert volt [9], hogy ez a két paraméter együttesen is korlátozható, minden rekurzívan felsorolható nyelv generálható hét vagy hat nemterminálist tartalmazó grammatikával aminek öt vagy hat szétszórt szabálya van. Ismert volt az is, hogy három nemterminális elegendő, egy viszont kevés minden rekurzívan felsorolható nyelv előállításához, [15, 16].

3. A szétszórt szövegfeltételekkel adott grammatikákban szükséges szétszórt szabályok és a nemterminálisok együttes számára, valamint pusztán a nemterminálisok számára is sikerült a korábbiaknál jobb korlátot adni:
 - (a) A szétszórt szövegfeltételekkel adott grammatikákban szükséges szétszórt szabályok és nemterminálisok együttes száma legfeljebb kettő és öt (3.4.1 tétel).
 - (b) A szétszórt szövegfeltételekkel adott grammatikákban szükséges nemterminálisok maximális száma kettő (3.4.2 tétel). Ez az eredmény optimális.

A párhuzamos, kommunikáló grammatikarendszerekkel kapcsolatban, a visszatérő esetben már korábban ismert, Turing gépek szimulációjára szolgáló technikát átültettük a nem visszatérő rendszerekre is, ami lehetőséget adott az ún. n számlálós automaták számításainak közvetlen szimulációjára. Bevezettük továbbá a klaszterezett PC grammatikarendszer fogalmát, megvizsgáltuk a komponensek előre definiált és a működés során rögzített (statikus), valamint a számítás során folyamatosan változó (dinamikus) klaszterekbe szervezésének lehetőségeit. Mindkét említett technika hozzájárult a PC grammatikarendszerek leírási bonyolultságának csökkentéséhez.

4. A nem visszatérő PC grammatikarendszerek leírási bonyolultsága a visszatérő rendszerekhez hasonlóan alacsony paraméterekkel korlátozható:
 - (a) A nem visszatérő rendszerek számítási ereje megegyezik a Turing gépek erejével abban az esetben is, ha produkciós szabályainak alakja egyszerű, komponenseinek száma pedig alacsony (4.2.4 tétel, 4.2.5 korollárium).
 - (b) A nem visszatérő PC grammatikarendszereknek sincs szükségük több komponensre maximális számítási erejük eléréséhez mint amennyire jelenlegi ismereteink szerint a visszatérő rendszereknek van (4.3.2 korollárium, 4.3.3 tétel).
5. A nem visszatérő PC grammatikarendszerekben zajló kommunikáció a komponensek klaszterekbe szervezésével egyszerűsíthető:
 - (a) A komponensek előre rögzített, statikus klaszterekbe sorolásakor a klaszterek számának csökkenésével (azaz az egy klaszterhez tartozó komponensek számának növekedésével) párhuzamosan növekszik a szükséges komponensek száma, de a maximális számítási erő eléréséhez a klaszterezett esetben sem feltétlenül kell a komponensek össz-számának növekednie (4.3.1 és 4.3.3 tétel).
 - (b) A komponensek klaszterezésének dinamikus, a működés során folyamatosan változó szervezésével a szükséges klaszterek száma tovább csökkenthető, miközben a komponensek száma az előre rögzített klaszterezéshez képest csak kisebb mértékben növekszik (4.3.1 és 4.3.3 tétel).

A minimális méretű szabályokat használó symport/antiport rendszerekről ismert volt, hogy négy membránnal képesek minden rekurzívan felsorolható számhalmazt előállítani, [10]. A korábban is használt bizonyítási módszert

követve, de új szimulációs technika alkalmazásával sikerült ezt a korlátot lejjebb szorítani.

6. Minden symport/antiport rendszerrel elvégezhető számítás elvégezhető minimális méretű szabályokat alkalmazó, legfeljebb három membránból álló rendszerrel is (5.2.1 tétel).

A P kolóniák esetén egy általunk bevezetett, korábban máshol már sikeresen alkalmazott bizonyítási technika segítségével sikerült a rendszert alkotó cellák számát és az egyes cellákhoz rendelt programok számát együttesen is korlátok közé szorítani. A maximum egyetlen objektumot tartalmazó (tehát minimális méretű) cellákból álló rendszerekről ismert volt, hogy egy viszonylag bonyolultnak tekinthető „érzékelő” szabály (checking rule) segítségével ezek az egyszerű építőelemek is képesek maximális számítási erő kifejtésére [1]. Megmutattuk, hogy erejük akkor sem csökken, ha nem engedjük meg az érzékelő szabályok használatát.

7. A P kolóniát alkotó cellák száma, az egyes cellákhoz rendelt programok száma, valamint az egyes cellákhoz rendelt programok bonyolultsága is korlátok köré szorítható:
 - (a) A P kolóniák celláinak és a cellákhoz rendelt programoknak a száma együttesen is korlátozható, a szükséges programok jelenleg ismert száma bonyolultságuk növelésével illetve érzékelő szabályok használatával csökken (5.3.3, 5.3.4, 5.3.5, 5.3.6 tétel).
 - (b) A két objektumot tartalmazó cellákból álló rendszerek egyszerűsített, ún. beszűrő/törlő (insertion/deletion) programok használata esetén is megőrzik maximális számítási erejüket (5.3.7 tétel).
 - (c) Az egy objektumot tartalmazó cellákból, azaz a lehető legegyszerűbb méret bonyolultságú cellákból álló rendszerek akkor is megőrzik maximális számítási erejüket, ha nem használnak érzékelő szabályokat (5.3.8 tétel).

4. Az eredmények hasznosítása

A disszertációban bemutatott eredmények új fogalmakkal, technikákkal bővítették a formális nyelvek elméletének eszköztárát, melyek a klasszikusnak tekinthető területek mellett a nem-hagyományos számítások, a nem-klasszikus számítási modellek vizsgálata szempontjából is érdekesek.

A külső kontroll-mechanizmusokkal vezérelt grammatikák leírási bonyolultságának vizsgálata közben kidolgozott technikák alkalmasak lehetnek más, hasonló típusú rendszerek vizsgálatára is. Különösen igaz ez a megállapítás a szétszórt szövegfeltételekkel adott grammatikákban szükséges nemterminálisok számának vizsgálatához használt szimulációs technikával kapcsolatban. A bizonyítás során használt konstrukció lényege, hogy a sikeres levezetések során megőrződő invariáns tulajdonság megléte biztosítja, hogy a szimulációt megkezdő produkciós szabály csak a levezetés első lépésében és csak egyetlen alkalommal használható, kiküszöbölve így a hasonló konstrukciókban általában szükséges elkülönített kezdő nemterminális szimbólum használatát. A technika hátránya, hogy jelenlegi formájában erősen támaszkodik a szétszórt szövegfeltételekkel adott grammatikákkal végzett levezetések speciális tulajdonságaira, de ez pontosabb megfontolások segítségével a későbbiekben talán kiküszöbölhető.

Hasonlóan az előbbiekhez, a P kolóniák méretével kapcsolatos paraméterek együttes korlátozásához használt univerzális regisztergép szimulációs technika is alkalmazható hasonló típusú szimbólumfeldolgozó rendszerek leírási bonyolultságával kapcsolatos vizsgálatok során. Az eljárás lényege, hogy amennyiben sikerül a [13] által leírt konkrét, kevés regiszterrel és kevés utasítással bíró univerzális regisztergép szimulációját valamilyen számítási modell segítségével megoldani, akkor automatikusan korlátokat kapunk a modell leírási bonyolultságával kapcsolatos paraméterekre, melyek a modell sajátosságait kihasználva, a szimuláció ésszerűsítésével általában tovább csökkenthetők.

A disszertáció „végleges”, egyes területeket lezáró eredményeket is tartalmaz. Ilyen a szétszórt szövegfeltételekkel adott grammatikák nemterminális számára vonatkozó optimális korlát, illetve a P kolóniák egyetlen objektumot tartalmazó variánsának számítási erejére vonatkozó tétel. A fentiekben vázolt, más területeken is hasznosítható általános technikák mellett ezek az eredmények is hozzájárulnak a formális nyelvek, illetve a nem-hagyományos számítások elméletének fejlődéséhez.

5. A disszertáció alapjául szolgáló publikációk

1. E. Csuhaj-Varjú, M. Margenstern, and Gy. Vaszil. P colonies with a bounded number of cells and programs. In H. J. Hoogeboom, Gh. Păun, G. Rozenberg, and A. Salomaa, editors, *Membrane Computing, 7th International Workshop, WMC 2006, Leiden, The Netherlands, July 17-21, 2006, Revised, Selected, and Invited Papers*, volume 4361 of *Lecture Notes in Computer Science*, pages 352–366. Springer, 2006.
2. E. Csuhaj-Varjú and Gy. Vaszil. On the number of components and clusters of non-returning parallel communicating grammar systems. In M. Holzer, M. Kutrib, and G. Pighizzini, editors, *Descriptive Complexity of Formal Systems. 13th International Workshop, DCFS 2011. Giessen/Limburg, Germany, July 2011. Proceedings*, volume 6808 of *Lecture Notes in Computer Science*, pages 121–134, Berlin Heidelberg, 2011. Springer-Verlag
3. E. Csuhaj-Varjú, Gy. Vaszil. Scattered context grammars generate any recursively enumerable language with two nonterminals. *Information Processing Letters*, 110(20):902–907, 2010.
4. E. Csuhaj-Varjú, Gy. Vaszil. On the descriptive complexity of context-free non-returning PC grammar systems. *Journal of Automata, Languages and Combinatorics*, 15(1–2):91–105, 2010.
5. Gy. Vaszil. On the nonterminal complexity of tree controlled grammars. In Henning Bordihn, Martin Kutrib, and Bianca Truthe, editors, *Languages Alive*, volume 7300 of *Lecture Notes in Computer Science*, pages 265–272. Springer, Berlin Heidelberg, 2012.
6. Gy. Vaszil. On the descriptive complexity of some rewriting mechanisms regulated by context conditions. *Theoretical Computer Science*, 330(2):361–373, 2005.
7. Gy. Vaszil. On the size of P systems with minimal symport/antiport. In G. Mauri, Gh. Păun, M. J. Pérez-Jiménez, G. Rozenberg, and A. Salomaa, editors, *Membrane Computing, 5th International Workshop, WMC 2004, Milan, Italy, June 14-16, 2004, Revised Selected and Invited Papers*, volume 3365 of *Lecture Notes in Computer Science*, pages 404–413. Springer, 2005.

8. L. Ciencialová, E. Csuhaj-Varjú, A. Kelemenová, and Gy. Vaszil. Variants of P colonies with very simple cell structure. *International Journal of Computers Communication and Control*, 4(3):224–233, 2009.

A t  zisekben hivatkozott irodalom

- [1] L. Cienciala, L. Ciencialov  , and Kelemenov  . A. On the number of agents in P colonies. In G. Eleftherakis, P. Kefalas, Gh. P  un, G. Rozenberg, and A. Salomaa, editors, *Membrane Computing. 8th International Workshop, WMC 2007. Thessaloniki, Greece, June 25-28, 2007. Revised Selected and Invited Papers*, volume 4860, pages 193–208, Berlin-Heidelberg, 2007. Springer.
- [2] E. Csuhaj-Varj  , J. Dassow, J. Kelemen, and Gh. P  un. *Grammar Systems: A Grammatical Approach to Distribution and Cooperation*, volume 5 of *Topics in Computer Mathematics*. Gordon and Breach Science Publishers, 1994.
- [3] E. Csuhaj-Varj  , Gh. P  un, and Gy. Vaszil. PC grammar systems with five context-free components generate all recursively enumerable languages. *Theoretical Computer Science*, 299(1-3):785–794, 2003.
- [4] K.   ulik II and H. Maurer. Tree controlled grammars. *Computing*, 19:129–139, 1977.
- [5] J. Dassow and G. P  un. *Regulated Rewriting in Formal Language Theory*. Springer, Berlin, 1989.
- [6] J. Dassow, G. P  un, and A. Salomaa. Grammars with controlled derivations. In A. Salomaa and G. Rozenberg, editors, *Handbook of Formal Languages*, volume II, pages 101–154. Springer, Berlin Heidelberg, 1997.
- [7] J. Dassow, Gh. P  un, and G. Rozenberg. Grammar systems. In A. Salomaa and G. Rozenberg, editors, *Handbook of Formal Languages*, volume II, chapter 4, pages 155–213. Springer-Verlag, Berlin-Heidelberg, 1997.
- [8] S. Dumitrescu. Non-returning PC grammar systems can be simulated by returning systems. *Theoretical Computer Science*, 165:463–474, 1996.
- [9] H. Fernau and A. Meduna. A simultaneous reduction of several measures of descriptive complexity in scattered context grammars. *Information Processing Letters*, 86:235–240, 2003.

- [10] P. Frisco. About p systems with symport/antiport. In *Second Brainstorming Week in Membrane Computing. Sevilla, February 2-7, 2004*, number 01/2004 in Technical Reports of the Research Group in Natural Computing, pages 224–236, Sevilla, 2004. University of Sevilla.
- [11] S. A. Greibach and J. E. Hopcroft. Scattered context grammars. *Journal of Computer and System Sciences*, 3:232–247, 1969.
- [12] J. Kelemen, A. Kelemenová, and Gh. Păun. Preview of p colonies: A biochemically inspired computing model. In M. Bedau, P. Husbands, T. Hutton, S. Kumar, and H. Suzuki, editors, *Workshop and Tutorial Proceedings. Ninth International Conference on the Simulation and Synthesis of Living Systems (Alife IX)*, pages 82–86, Boston Mass., 2004.
- [13] I. Korec. Small universal register machines. *Theoretical Computer Science*, 168(2):267–301, 1996.
- [14] N. Mandache. On the computational power of context-free PC grammar systems. *Theoretical Computer Science*, 237(1-2):135–148, 2000.
- [15] A. Meduna. Generative power of three-nonterminal scattered context grammars. *Theoretical Computer Science*, 246:423–427, 2000.
- [16] A. Meduna. Terminating left-hand sides of scattered context productions. *Theoretical Computer Science*, 237:423–427, 2000.
- [17] A. Meduna and A. Gopalaratnam. On semi-conditional grammars with productions having either forbidding or permitting conditions. *Acta Cybernetica*, 11:307–323, 1994.
- [18] A. Meduna and M. Švec. Reduction of simple semi-conditional grammars with respect to the number of conditional productions. *Acta Cybernetica*, 15:353–360, 2002.
- [19] Gh. Păun, G. Rozenberg, and A. Salomaa, editors. *The Oxford Handbook of Membrane Computing*. Oxford University Press, 2010.
- [20] A. Păun and Gh. Păun. The power of communication: P systems with symport/antiport. *New Generation Computing*, 20(3):295–306, 2002.
- [21] Gh. Păun. *Membrane Computing: An Introduction*. Natural Computing Series. Springer, Berlin, 2002.

- [22] S. Turaev, J. Dassow, and M. Selamat. Language classes generated by tree controlled grammars with bounded nonterminal complexity. In M. Holzer, M. Kutrib, and G. Pighizzini, editors, *Descriptional Complexity of Formal Systems, 13th International Workshop, DCFS 2011, Gießen-Limburg, Germany, July 2011, Proceedings.*, volume 6808 of *Lecture Notes in Computer Science*, pages 289–300, Berlin Heidelberg, 2011. Springer.
- [23] Gy. Vaszil. On simulating non-returning PC grammar systems with returning systems. *Theoretical Computer Science*, 209:319–329, 1998.
- [24] Gy. Vaszil. Non-returning PC grammar systems generate any recursively enumerable language with eight context-free components. *Journal of Automata, Languages and Combinatorics*, 12(1-2):307–315, 2007.