**Formal Methods in Software Engineering** - Informatică, an 3, sem 2

An univ. 2024-2025

Curs: Florentin Ipate

Laborator: Mihail Pleșa

Invited speaker: Marian Gheorghe (University of Bradford).

1. **Obiective**: Asimilarea principalelor concepte si tehnici de formal modelling and verification, model-based testing, membrane computing și quantum computing.
2. **Cerinte**

Notarea se va face pe baza de proiect. Proiectele se vor efectua in echipe (dimensiunea maxima a echipei pentru fiecare tema este specificata mai jos) si vor fi prezentate la curs sau la laborator, in functie de tema aleasa (conform informațiilor de mai jos), in saptamanile 5-10 conform programarii stabilite cu cadrul didactic de curs/laborator, astfel incat numarul de studenti care prezinta in cadrul unui curs/laborator sa nu depaseasca 10. Tema proiectului va fi aleasa din lista de mai jos. Fiecare tema din lista poate fi aleasa de cel mult 2 echipe.

Prezentarea va fi sub forma de slide-uri, insotite de demo-uri. La prezentare este necesara prezenta intregii echipe, fiecare student descriind principala sa contributie la proiect. Timpul alocat fiecarui student este de 10 minute. Atunci cand echipa considera ca nu toti membrii echipei au avut o contributie egala la realizarea proiectului, se va indica, in procente, contributia estimata a fiecaruia.

1. **Teme proiecte**
2. **Formal Modelling and Verification** – prezentarea acestor teme se va face la curs
3. **Modelare si analiza cu Event-B, Rodin**

* Prezentati capabilitatile de modelare si analiza ale limbajului Event-B: evenimente, rafinare, demonstrare automata a proprietatilor.
* Ilustrati aceste capabilitati folosind unul sau mai multe exemple create de echipa.

Marime maxima echipa: 3 studenti

Bibliografie: [1,2]

1. **Model checking cu Event-B, Rodin si Pro B**

* Prezentati abordarea de model checking folosind Event-B, Rodin si Pro B descrisa in [3].
* Ilustrati aceasta abordare folosind unul sau mai multe exemple create de echipa (diferite de exemplul din articolul [3]).

Marime maxima echipa: 3 studenti

Bibliografie: [3,4]

1. **Model checking cu NuSMV**

* Prezentati pe scurt logica LTL si logica CTL.
* Ilustrati capabilitatile de model checking (verificare de proprietati LTL si CTL) folosind unul sau mai multe exemple create de echipa.

Marime maxima echipa: 3 studenti

Bibliografie: [5,6]

1. **Model-based Testing** – prezentarea acestor teme se va face la curs
2. **Model based testing cu JSXM**

* Prezentati limbajul de modelare bazat pe stream X-machines precum si principalele facilitati oferite de utilitarul JSXM: animarea modelului, generarea testelor pe baza modelului de forma stream X-machine, implementarea testelor in JUnit.
* Ilustrati aceste capabilitati folosind unul sau mai multe exemple create de echipa.

Marime maxima echipa: 3 studenti

Bibliografie: [7,8,9]

Observatie: pentru instalarea tool-ului se va consulta documentul „JSXM installation instructions”.

1. **Search based testing pentru EFSM**

* Prezentati metoda de generare de date de test prezentata in [10].
* Scrieti un program care implementeaza metoda de generare de date de test prezentata in [10] si ilustrati functionarea acesteia pe un exemplu.

Marime maxima echipa: 2 studenti

Bibliografie: [10]

1. **GUI testing cu Fastbot2**

* Prezentati principalele facilitati oferite de utilitarul Fastbot2: construirea un model probabilistic și strategii de testare ghidată bazată pe model (îmbunătățită printr-un algoritm de reinforcement learning).
* Ilustrati aceste capabilitati folosind unul sau mai multe exemple create de echipa.

Marime maxima echipa: 3 studenti

Bibliografie: [11,12]

1. **Model based testing cu GraphWalker**

* Prezentati principalele facilitati oferite de utilitarul GraphWalker: generarea testelor din modelul masinilor cu stari finite (FSM) folosind algoritmi de traversare precum A\* sau parcurgere random cu diverse criterii de acoperire (state, edge, requirement).
* Ilustrati aceste capabilitati folosind unul sau mai multe exemple create de echipa.

Marime maxima echipa: 2 studenti

Bibliografie: [13,14]

1. **Membrane Computing Applications** - prezentarea acestor teme se va face la curs
2. **P-Lingua and Mecosim**

* Prezentati limbajul P-Lingua si ilustrati folosirea acestuia pe un exemplu creat de echipa.
* Utilizati MeCoSim pentru simularea si analiza exemplului creat anterior.

Marime maxima echipa: 3 studenti

Bibliografie: [15,16,17]

1. **Modelling and simulation of tissue P systems**

* Prezentati sintaxa limbajului P-Lingua pentru tissue P systems si ilustrati folosirea acestuia pe un exemplu creat de echipa.
* Prezentati intuitiv algoritmul de simulare a tissue P systems si ilustrati functionarea acestuia pe exemplul dat.

Marime maxima echipa: 3 studenti

Bibliografie: [17,18]

1. **Modelling and simulation of spiking neural P systems**

* Prezentati sintaxa limbajului P-Lingua pentru spiking neural P systems si ilustrati folosirea acestuia pe un exemplu creat de echipa.
* Prezentati algoritmul de simulare a spiking neural P systems si ilustrati functionarea acestuia pe exemplul dat.

Marime maxima echipa: 3 studenti

Bibliografie: [17,19]

1. **kP Systems si KPWorkbench**

* Prezentati modelul kP System si ilustrati acest formalism printr-un exemplu creat de echipa.
* Prezentati facilitatile KPWorkBench (modelare, simulare si verificare formala) pe exemplul anterior.

Marime maxima echipa: 3 studenti

Bibliografie: [20,21,22]

1. **Modeling and Verification of P Systems Using Rodin and ProB**

* Ilustrati printr-un exemplu reprezentarea in Event-B a unui P system.
* Ilustrati pe exemplul anterior verificarea formala folosind ProB.

Marime maxima echipa: 3 studenti

Bibliografie: [23,24,25]

1. **Formal verification of P Systems Using NuSMV**

* Ilustrati printr-un exemplu reprezentarea in SMV a unui transformation-communication P system.
* Ilustrati pe exemplul anterior verificarea formala folosind NuSMV.

Marime maxima echipa: 3 studenti

Bibliografie: [26,27]

1. **Model checking based test generation from P systems**

* Prezentati criteriile de acoperire rule coverage si context-dependent rule coverage si ilustrati aceste tipuri de acoperire cu un exemplu creat de echipa.
* Aratati cum pot fi generate teste pentru criteriile de acoperire mai sus mentionate si generati teste pentru exemplul de mai sus folosind NuSMV.

Marime maxima echipa: 3 studenti

Bibliografie: [27,28]

1. **Membrane Computing Applications** - prezentarea acestor teme se va face la laborator
2. **Spiking Neural Networks (SNNs) for classification**

* Prezentați conceptul de SNN.
* Implementați un PoC folosind un simulator pentru SNN clasificarea dataset-ului MNIST.

Mărime maxima echipa: 5 studenti

Bibliografie: [29, 30, 31, 32, 33]

Resurse: <https://github.com/jeshraghian/snntorch>, <https://www.kaggle.com/code/dlarionov/mnist-spiking-neural-network>

1. **Membrane algorithms**

* Prezentați algoritmii de mebrane computing precum și aplicațiile acestora.
* Implementați un PoC pentru a studia experimental eficiența algoritmilor de membrane computing în comparație cu abordările clasice.

Mărime maxima echipa: 5 studenti

Bibliografie: [34, 35, 36, 37]

1. **Neural cryptography**

* Prezentați conceptul de neural cryptography.
* Integrați un protocol de key agreement bazat pe sincronizare neuronală în implementarea unui sistem open-source (wireguard, signal, TLS, etc.)

Mărime maxima echipa: 5 studenti

Bibliografie: [38, 39, 40, 41, 42]

1. **Membrane computing for cryptanalysis**

* Prezentați algoritmii de membrane computing folosiți pentru criptanaliză.
* Implementați simularea unui atac de criptanaliză.

Mărime maxima echipa: 5 studenti

Bibliografie: [43, 44, 45, 46, 47]

1. **Adverarial neural cryptography**

* Prezentați conceptul de adversarial cryptography.
* Implementați un algoritm de criptare simetric bazat pe neural cryptography și integrați-l într-un sistem open-source (wireguard, signal, TLS, etc)

Mărime maxima echipa: 5 studenti

Bibliografie: [48, 49, 50]

1. **Spiking Neural P systems on CUDA**
   * Prezentați conceptul de spiking neural P systems
   * Implementați un simulator de SN P systems folosind CUDA.

Mărime maximă echipă: 5 studenți

Bibliografie: [51, 52, 53, 54, 55, 56]

1. **Quantum Computing**  - prezentarea acestor teme se va face la laborator
2. **Quantum factorization**

* Prezentați algoritmul lui Shor
* Implementați algoritmul și demonstrați factorizarea unui număr folosind un sistem de calcul real (IBM Q)

Mărime maxima echipa: 5 studenti

1. **Quantum searching**

* Prezentați algoritmul lui Grover
* Implementați algoritmul și demonstrați căutarea unui număr într-un șir folosind un sistem de calcul real (IBM Q)

Mărime maxima echipa: 5 studenti

1. **Quantum ML** 
   * Prezentați conceptele de quantum machine learning
   * Implementați un algoritm de clasificare și demontrați performanțele acestuia pe un dataset simplu folosind un sistem de calcul real (IBM Q)

Mărime maxima echipa: 5 studenti

**Resurse pentru 21, 22, 23:**

* <https://qiskit.org/textbook-beta/course/introduction-course>
* [💻 Qiskit Foundations - Coding with Qiskit Season 1](https://www.youtube.com/playlist?list=PLOFEBzvs-Vvp2xg9-POLJhQwtVktlYGbY)
* <https://medium.com/qiskit/qiskit-and-its-fundamental-elements-bcd7ead80492>
* <https://qiskit.org/documentation/tutorials/circuits/1_getting_started_with_qiskit.html>

1. **Bibliografie**
2. <http://www.event-b.org/>
3. Jean-Raymond Abrial. Modeling in Event-B: System and Software Engineering https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=0861902a5167a3f98b4fe2532f3a83d6aa4a43d5
4. Adrian Turcanu, Talal Shaikh and Cristina Nicoleta Mazilu, On Model Checking of a Robotic Mechanism. <https://scholars.direct/Articles/robotics/jra-4-018.pdf>
5. <https://www3.hhu.de/stups/prob/>
6. <https://www.cs.utexas.edu/users/moore/acl2/seminar/2010.05-19-krug/slides.pdf>
7. <https://nusmv.fbk.eu/>
8. Florentin Ipate, Mike Holcombe: Specification and testing using generalized machines: a presentation and a case study, Software Testing, Verification and Reliability, 8, 61-81 (1998) <http://www.ifsoft.ro/~florentin.ipate/publications/STVR98%20-%20Specification%20and%20testing%20using%20generalised%20machines.pdf>
9. <http://www.jsxm.org/>
10. Dimitris Dranidis, Konstantinos Bratanis, Florentin Ipate: JSXM: a tool for automated test generation, in SEFM, 352-366 (2012).
11. Raluca Lefticaru, Florentin Ipate: Automatic State-Based Test Generation Using Genetic Algorithms, in SYNASC, 188-195 (2007). <http://www.ifsoft.ro/~florentin.ipate/publications/SYNASC%202007%20Automatic%20State-Based%20Test%20Generation%20Using%20Genetic%20Algorithms.pdf>
12. <https://github.com/bytedance/Fastbot_Android>
13. <https://tingsu.github.io/files/ASE22-industry-Fastbot.pdf>
14. <https://graphwalker.github.io/>
15. Muhammad Nouman Zafar, Wasif Afzal, Eduard Enoiu, Athanasios Stratis, Aitor Arrieta, Goiuria Sagardui: Model-Based Testing in Practice: An Industrial Case Study using GraphWalker. ISEC 2021: 5:1-5:11 <http://ebiltegia.mondragon.edu/xmlui/bitstream/handle/20.500.11984/5411/Model-Based%20Testing%20in%20Practice_An%20Industrial%20Case%20%20Study%20using%20GraphWalker.pdf?sequence=1&isAllowed=y>
16. A P-Lingua Programming Environment for Membrane Computing. Workshop on Membrane Computing 2008: 187-203. <https://idus.us.es/bitstream/handle/11441/68600/1/p-lingua-WMC9.pdf?sequence=1>
17. Ignacio Pérez-Hurtado, Luis Valencia-Cabrera, Mario J. Pérez-Jiménez, Maria Angels Colomer, Agustin Riscos-Núñez: MeCoSim: A general purpose software tool for simulating biological phenomena by means of P systems. BIC-TA 2010: 637-643. <http://www.p-lingua.org/mecosim/doc/_downloads/mecosim.pdf>
18. <https://github.com/RGNC/plingua>
19. Miguel A. Martínez-del-Amor, Ignacio Pérez-Hurtado, Mario J. Pérez-Jiménez, Agustin Riscos-Núñez: A P-Lingua based simulator for tissue P systems. J. Log. Algebraic Methods Program. 79(6): 374-382 (2010) <https://www.sciencedirect.com/science/article/pii/S1567832610000135/pdf?md5=5145407652ddc0bb0809ce8f3ea46cec&pid=1-s2.0-S1567832610000135-main.pdf>
20. Luis F. Macías-Ramos, Ignacio Pérez-Hurtado, Manuel García-Quismondo, Luis Valencia-Cabrera, Mario J. Pérez-Jiménez, Agustin Riscos-Núñez: A P-Lingua Based Simulator for Spiking Neural P Systems. Int. Conf. on Membrane Computing 2011: 257-281 <https://idus.us.es/bitstream/handle/11441/79580/Mac%C3%ADasRamos2012_Chapter_APLinguaBasedSimulatorForSpiki.pdf>
21. S. Konur, L. Mierla, F. Ipate, M. Gheorghe: kpworkbench: A software suit for membrane systems, SoftwareX, 11: 100407, 2020. <https://www.ifsoft.ro/~florentin.ipate/publications/SoftwarexKMIG.pdf>
22. M. Gheorghe, F. Ipate, L. Mierla, S. Konur: kPWorkbench: A Software Framework for Kernel P Systems, BWMC 2015, 179-194, 2015. <https://idus.us.es/xmlui/bitstream/handle/11441/33048/179_bwmc15.pdf?sequence=1&isAllowed=y>
23. <https://github.com/Kernel-P-Systems/kPWorkbench>
24. F. Ipate, A. Turcanu: Modeling, Verification and Testing of P Systems Using Rodin and ProB, in Ninth Brainstorming Week on Membrane Computing, 209-220, Fenix Editora, Sevilla, 2011. <https://www.ifsoft.ro/~florentin.ipate/publications/BWMC%202011%20Verification%20and%20Testing%20of%20P%20Systems%20Using%20Rodin%20and%20ProB.pdf>
25. <http://www.event-b.org/>
26. <https://prob.hhu.de/>
27. M. Gheorghe, F. Ipate, R. Lefticaru, C.Dragomir: An integrated approach to P systems formal verification, in Int. Conf. on Membrane Computing (CMC 2010), Jena, Germany, August 24-27, 2010. Revised Selected Papers, LNCS 6501 Springer, 226-239, 2011. <https://www.ifsoft.ro/~florentin.ipate/publications/LNCS%20CMC%202010%20An%20Integrated%20Approach%20to%20P%20Systems%20Formal%20Verification.pdf>
28. <https://nusmv.fbk.eu/>
29. R. Lefticaru, F. Ipate, M. Gheorghe: Model Checking Based Test Generation from P Systems Using P-Lingua, Romanian Journal of Information Science and Technology, 13(2), 153-168, 2010. <https://www.ifsoft.ro/~florentin.ipate/publications/ROMJIST%202010%20Model%20Checking%20Based%20Test%20Generation%20from%20P%20Systems%20Using%20P-Lingua.pdf>
30. Lee, J.H., Delbruck, T. and Pfeiffer, M., 2016. Training deep spiking neural networks using backpropagation. Frontiers in neuroscience, 10, p.508.  
    <https://www.frontiersin.org/articles/10.3389/fnins.2016.00508/full>
31. Tavanaei, A., Ghodrati, M., Kheradpisheh, S.R., Masquelier, T. and Maida, A., 2019. Deep learning in spiking neural networks. Neural networks, 111, pp.47-63.  
    <https://arxiv.org/pdf/1804.08150>
32. Vaila, R., Chiasson, J. and Saxena, V., 2019. Deep convolutional spiking neural networks for image classification. arXiv preprint arXiv:1903.12272.  
    <https://arxiv.org/pdf/1903.12272>
33. Zheng, H., Wu, Y., Deng, L., Hu, Y. and Li, G., 2021, May. Going deeper with directly-trained larger spiking neural networks. In Proceedings of the AAAI conference on artificial intelligence (Vol. 35, No. 12, pp. 11062-11070).  
    <https://ojs.aaai.org/index.php/AAAI/article/download/17320/17127>
34. Diehl, P.U. and Cook, M., 2015. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. Frontiers in computational neuroscience, 9, p.99.  
    <https://www.frontiersin.org/articles/10.3389/fncom.2015.00099/full>
35. Zhang, G., Liu, C., Gheorghe, M. and Ipate, F., 2009, October. Solving satisfiability problems with membrane algorithms. In 2009 Fourth International on Conference on Bio-Inspired Computing (pp. 1-8). IEEE.  
    <http://www.ifsoft.ro/~florentin.ipate/publications/BIC-TA%202009%20Solving%20Satisfiability%20Problems%20with%20Membrane%20Algorithms.pdf>
36. Nicolescu, R., Dinneen, M.J., Cooper, J., Henderson, A. and Liu, Y., 2022. Logarithmic sat solution with membrane computing. Axioms, 11(2), p.66.  
    <https://www.mdpi.com/2075-1680/11/2/66/pdf>
37. Jimen, Y. and Fujiwara, A., 2018. An asynchronous P system with branch and bound for solving the satisfiability problem. International Journal of Networking and Computing, 8(2), pp.141-152.  
    <https://www.jstage.jst.go.jp/article/ijnc/8/2/8_141/_pdf>
38. Adorna, H.N., Pan, L. and Song, B., 2018. On distributed solution to SAT by membrane computing. INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL, 13(3), pp.303-320.  
    <https://univagora.ro/jour/index.php/ijccc/article/download/3217/1204>
39. Plesa, M.I., Gheoghe, M., Ipate, F. and Zhang, G., 2022. A key agreement protocol based on spiking neural P systems with anti-spikes. Journal of Membrane Computing, 4(4), pp.341-351.  
    <http://www.ifsoft.ro/~florentin.ipate/publications/KeyAgreementProtocol.pdf>
40. Rosen-Zvi, M., Klein, E., Kanter, I. and Kinzel, W., 2002. Mutual learning in a tree parity machine and its application to cryptography. Physical Review E, 66(6), p.066135.  
    <https://arxiv.org/pdf/cond-mat/0209234>
41. Volkmer, M. and Wallner, S., 2005. Tree parity machine rekeying architectures. IEEE Transactions on Computers, 54(4), pp.421-427.  
    <https://arxiv.org/pdf/cs/0502062>
42. Jeong, S., Park, C., Hong, D., Seo, C. and Jho, N., 2021. Neural cryptography based on generalized tree parity machine for real-life systems. Security and communication networks, 2021, pp.1-12.  
    <https://www.hindawi.com/journals/scn/2021/6680782/>
43. Stypiński, M. and Niemiec, M., 2022. Synchronization of tree parity machines using nonbinary input vectors. IEEE Transactions on Neural Networks and Learning Systems.  
    <https://ieeexplore.ieee.org/iel7/5962385/6104215/09795149.pdf>
44. Wang, H., Zhou, K., Zhang, G., Paul, P., Duan, Y., Qi, H. and Rong, H., 2020. Application of Weighted Spiking Neural P Systems with Rules on Synapses for Breaking RSA Encryption. Int. J. Unconv. Comput., 15(1-2), pp.37-58.  
    <https://researchspace.auckland.ac.nz/bitstream/handle/2292/45075/530ACMC2018.pdf?sequence=1#page=194>
45. Vasile, R., Gheorghe, M. and Niculescu, I.M., 2023. Breaking RSA Encryption Protocol with Kernel P Systems.  
    <https://www.researchsquare.com/article/rs-2684530/latest.pdf>
46. Mushtaq, M., Mukhtar, M.A., Lapotre, V., Bhatti, M.K. and Gogniat, G., 2020. Winter is here! A decade of cache-based side-channel attacks, detection & mitigation for RSA. Information Systems, 92, p.101524.  
    <https://www.sciencedirect.com/science/article/am/pii/S0306437920300338>
47. Yarom, Y., Genkin, D. and Heninger, N., 2017. CacheBleed: a timing attack on OpenSSL constant-time RSA. Journal of Cryptographic Engineering, 7, pp.99-112.  
    <https://eprint.iacr.org/2016/224.pdf>
48. Zaher, S., Badr, A., Farag, I. and Abd Elmageed, T., 2012. Performance enhancement of RSA cryptography algorithm by membrane computing. Int J Adv Res Compt Sci Softw Eng, 2(9).  
    <https://www.researchgate.net/profile/Salah-Olaymi/publication/373776373_Performance_Enhancement_of_RSA_Cryptography_Algorithm_by_Membrane_Computing/links/64fc545dc5dd8170a0837da3/Performance-Enhancement-of-RSA-Cryptography-Algorithm-by-Membrane-Computing.pdf>
49. Abadi, M. and Andersen, D.G., 2016. Learning to protect communications with adversarial neural cryptography. arXiv preprint arXiv:1610.06918.  
    <https://arxiv.org/pdf/1610.06918>
50. Coutinho, M., de Oliveira Albuquerque, R., Borges, F., Garcia Villalba, L.J. and Kim, T.H., 2018. Learning perfectly secure cryptography to protect communications with adversarial neural cryptography. Sensors, 18(5), p.1306.  
    <https://www.mdpi.com/1424-8220/18/5/1306/pdf>
51. Meraouche, I., Dutta, S., Tan, H. and Sakurai, K., 2021. Neural networks-based cryptography: A survey. IEEE Access, 9, pp.124727-124740.  
    <https://ieeexplore.ieee.org/iel7/6287639/6514899/09527229.pdf>
52. Carandang, J.P., Villaflores, J.M.B., Cabarle, F.G.C., Adorna, H.N. and Martínez del Amor, M.Á., 2017. CuSNP: Spiking neural P systems simulators in CUDA. Romanian Journal of Information Science and Technology (ROMJIST), 20 (1), 57-70.  
    <https://idus.us.es/bitstream/handle/11441/106266/CuSNP%20Spiking%20neural%20P%20systems%20simulators%20in%20CUDA.pdf?sequence=1>
53. Cabarle, F.G.C., Adorna, H. and Martínez, M.A., 2012. A spiking neural P system simulator based on CUDA. In Membrane Computing: 12th International Conference, CMC 2011, Fontainebleau, France, August 23-26, 2011, Revised Selected Papers 12 (pp. 87-103). Springer Berlin Heidelberg.  
    <https://idus.us.es/bitstream/handle/11441/86848/A%20Spiking%20Neural%20P%20System%20Simulator.pdf?sequence=1>
54. Aboy, B.C.D., Bariring, E.J.A., Carandang, J.P., Cabarle, F.G.C., De La Cruz, R.T., Adorna, H.N. and Martínez-del-Amor, M.Á., 2019, July. Optimizations in cusnp simulator for spiking neural p systems on cuda gpus. In 2019 International Conference on High Performance Computing & Simulation (HPCS) (pp. 535-542). IEEE.  
    <https://idus.us.es/bitstream/handle/11441/106472/Optimizations%20in%20CuSNP%20Simulator%20for%20Spiking%20Neural%20P%20Systems%20on%20CUDA%20GPUs.pdf?sequence=1>
55. Carandang, J.P., Villaflores, J.M.B., Cabarle, F.G.C., Adorna, H.N. and Martínez del Amor, M.Á., 2016. Improving Simulations of Spiking Neural P Systems in NVIDIA CUDA GPUs: CuSNP. BWMC 2016: 14th Brainstorming Week on Membrane Computing: Sevilla, ETS de Ingeniería Informática, February 1-5 (2016), p 135-150.  
    <https://idus.us.es/bitstream/handle/11441/49802/1/135_cusnp-bwmc16.pdf?sequence=1>
56. Cabarle, F.G., Adorna, H., Martínez-del-Amor, M.A. and Pérez-Jiménez, M.J., 2011. Spiking neural P system simulations on a high performance GPU platform. In Algorithms and Architectures for Parallel Processing: 11th International Conference, ICA300 2011, Melbourne, Australia, October 24-26, 2011, Proceedings, Part II 11 (pp. 99-108). Springer Berlin Heidelberg.  
    <https://idus.us.es/bitstream/handle/11441/70521/1/f7a59986e9f120642d651165a0e1d29d4c1c.pdf?sequence=1>
57. Cabarle, F.G.C., Adorna, H. and Martinez–del–Amor, M.A., 2011, September. An improved GPU simulator for spiking neural P systems. In 2011 Sixth International Conference on Bio-Inspired Computing: Theories and Applications (pp. 262-267). IEEE.  
    <https://idus.us.es/bitstream/handle/11441/86890/An%20Improved%20GPU%20Simulator.pdf?sequence=1>