Master Thesis Opponent Report

Author: Parth Mittal
Title: Pseudorandom walks and chip firing games
Opponent: doc. Mgr. Zdeněk Dvořák, Ph.D.

Summary and evaluation

The first part of the thesis studies chip-firing games (where tokens are placed on vertices of a directed graph, and sent along the edges from vertices that have more tokens than their outdegree). It is shown by a relatively straightforward construction that these games can be used to simulate Turing machine, with the standard decidability and complexity consequences.

The second part considers a different chip-firing model, where at each vertex, a "rotor" fires the chips to outgoing edges in some given order. The goal here is to compare the distribution of chips on vertices with the one obtained by moving the chips at random. The thesis provides upper bounds on the difference between the distributions in some (quite general) special cases; this is a technically more involved part of the thesis, building upon the ideas developed by Kijima et al. Additionally, some simpler observations are made on the limit behavior.

The writing is quite clear, with just a couple of typos (diagonlizable, ...) Personally, I'd appreciate a more extensive introduction, with more background on the considered models and related topics. The current introductory sections are adequate, but (especially for a thesis) rather brief and do not do a great job in motivating the models and explaining the importance of the obtained results.

Overall, this is a quite nice work with some original results, without doubts matching the requirements for a master's thesis.

Remarks and questions

- In Theorem 1.3, why do you increase the depth by the factor of $\log s(n)$? Using the binary tree in the fan-out gadget seems unnecessary, to do a fan-out to k branches, it suffices to use a vertex of outdegree k with $c(v) = k - 1$.
- Lemma 1.4: For S', the outputs could also become strictly greater than 1.

- A somewhat inelegant aspect of Theorem 1.7 is that the constructed graph never stops firing, even if the machine stops. It seems that this could be improved along the following lines: Have a half-bit in the input/output that becomes 0 when a stopping state is reached, and thread this half-bit through the discard. This way when a stopping state is reached, the discard stops firing. Some additional fiddling may be needed to ensure that the whole graph actually stops firing at this point. If this approach works, it would presumably fix the problem that Theorem 1.9 is dependent on the firing strategy?

- A formal definition of the Propp machine is not given, and unfortunately one point is not quite clear from the informal description: Does the "rotor" reset to the first element of the permutation at the beginning of each time-step, or does it continue from the point where it fired last in the previous time-step? That is, it is not clear whether in the phrase "the i-th token ... overall", the word "overall" scopes to the current time-step or the whole process.

- "lowerbound" is usually not spelled as a single word

2.6.2021, Zdeněk Dvořák