Deterministic Random Walks on Infinite Grids

Diplomarbeit

zur Erlangung des akademischen Grades
Diplom-Mathematiker

Tobias Friedrich, M.Sc.
geb. am 28.07.1980 in Neustrelitz

Betreuer an der Friedrich-Schiller-Universität Jena:

PD Dr. rer. nat. habil. Harald Hempel

Betreuer am Max-Planck-Institut für Informatik, Saarbrücken:

Dr. rer. nat. habil. Benjamin Doerr

Jena, den 19.12.2005
Zusammenfassung


The rotor-router model is a simple deterministic analogue of random walk invented by Jim Propp. Instead of distributing chips to randomly chosen neighbors, it serves the neighbors in a fixed order. This thesis investigates how well this process simulates a random walk on an infinite two-dimensional grid. Independent of the starting configuration, at each time and on each vertex, the number of chips on this vertex deviates from the expected number of chips in the random walk model by at most a constant $c$. It is proved that $7.2 < c < 11.8$ in general. Surprisingly, these bounds depend on the order in which the neighbors are served. It is also shown that in a generalized setting, where one just requires that no neighbor gets more than $\Delta$ chips more than another, there is also such a constant $c'$ with $7.7\Delta < c' < 26.9\Delta$.

Acknowledgements

I would like to thank Dr. Benjamin Doerr for drawing my interest on deterministic random walks and for many long nights working together. I am also grateful to Dr. Harald Hempel for supervising this thesis.

I am thankful to the German National Merit Foundation (“Studienstiftung des deutschen Volkes”) and the Max Planck Society for financial support, good working conditions and travel allowance.
# Contents

1 Introduction 1  
1.1 Deterministic Random Walks .............................. 1  
1.2 Preliminaries and Notation ................................. 3  
1.3 Rotor Sequence ........................................... 6  
1.4 Related Work ............................................. 7  

2 The Basic Method 12  
2.1 Characterizing the Discrepancy .............................. 12  
2.2 Unimodality ............................................... 15  

3 Upper Bound 21  
3.1 Dependence on the Chosen Rotor Sequence .................... 21  
3.2 Convergence of the Series .................................. 24  
3.3 Computation ............................................... 25  

4 Lower Bound 31  
4.1 Arrow-forcing Theorem ...................................... 31  
4.2 Computation ............................................... 35  

5 Free Propp Machine 38  
5.1 Upper Bound ............................................... 38  
5.2 Lower Bound ............................................... 41  

6 Conclusions and Further Work 42  

Bibliography 44  

A Mathematical Basics 48
Chapter 1

Introduction

1.1 Deterministic Random Walks

Quasirandomness

Since antiquity, randomness and unpredictability were discussed in connection both with questions of free will and games of chance. In the 18th century randomness became very important in philosophy and science. However, nowadays true randomness is not available in computer simulations and also often undesired. Hence, a growing interest in “quasirandomness” is recently noticeable [6]. This is a deterministic progress specifically designed to give the same limiting behavior (of some quantities of interest) as a random process, but with faster convergence. Quasirandomness is not to be confused with “pseudorandomness”, a process that is supposed to be statistically indistinguishable from the truly random process it imitates. In contrast to this, quasirandomness will typically have many regularities that mark it as non-random.

Often, with a quasirandom simulation that focuses on reducing discrepancy, one can obtain much more precise estimates of the average case behavior of the system than via (purely random) Monte Carlo simulation. In other words, a quasirandom process gives deterministic error-bounds rather than (usually larger) confidence-intervals.
Random Walk

Given an arbitrary graph, a random walk of a chip is a path, which begins at a given starting point and chooses the next node with equal probability out of the set of its current neighbors. Random walks have been used to model a wide variety of processes in economics (share prices), physics (Brownian motion of molecules in gases and liquids), medicine (cascades of neurons firing in the brain), and mathematics (estimations and modeling of gambling).

Propp Machine

The Propp machine is the quasirandom analogue of random walk invented by Jim Propp. He called this the “Rotor router model”, while Joel Spencer [10] coined the term “Propp machine”. We will use the latter.

Instead of distributing chips to randomly chosen neighbors, it deterministically serves the neighbors in a fixed order by associating to each vertex a “rotor” pointing to one of its neighbors. The first chip reaching a vertex moves on in the direction the rotor is pointing, then the rotor direction is updated to the next direction in some cyclic ordering. The second chip is then sent in this new direction, the rotor is updated, and so on. This ensures the chips are distributed highly evenly among the neighbors.

The question is, how well the Propp machine resembles an expected random walk. A comparison can be drawn by contrasting different aspects. We focus on the single vertex discrepancy. That is, we distribute some finite number of chips arbitrarily on the vertices and let evolve this distribution with the chips following the rotors. One step consists of every chip incrementing and then following the rotor at the point it is on. Running in parallel Propp machine and expected random walk on the same initial configuration, we calculate the maximal deviation of the number of chips on each vertex. Jim Propp on the other hand, concentrates on the aggregation model, where each chip is content to stay put when they reach an uninhabited site (cf. Section 1.4).

On strong connected finite graphs the discrepancy on a single vertex is bounded by a constant. For arbitrary graphs there is not much known about the Propp machine. This is different for infinite grids $\mathbb{Z}^d$. There one can fix a certain order (cf. Section 1.3) in which the $2d$ neighbors are served (e.g., North, East, South, and West for $d = 2$). Astonishingly, Cooper and Spencer [10, 11] showed that this discrepancy is bounded by a constant $c_d$, which only depends on the dimension $d$. 
i.e., this constant is independent of the number of steps, the initial distribution of the chips, and the states of the rotors. Note, that this is only valid for “checkered” initial distributions, but this technicality we defer to Section 1.2. The constant discrepancy is particularly surprising when keeping in mind that in a single step the random walk splits \( n \) chips to its \( 2^d \) neighbors with a standard deviation of \( \sqrt{n} \). This shows that the Propp machine simulates a random walk extremely well. Together with Doerr and Tardos, they further analyzed the one-dimensional case in [8, 9]. They showed that for \( \mathbb{Z}^1 \) the actual constant is \( c_1 \approx 2.29 \). Further on, for intervals of length \( L \), the deviation between both models is only \( O(\log L) \) (instead of, e.g., \( 2.29L \)).

**Our Results**

We continue the work of Cooper et al. [8, 9] by regarding the single vertex discrepancy on the two-dimensional grid \( \mathbb{Z}^2 \). This is the most frequently considered infinite grid for random walks. In Chapter 3 we prove that independent of the initial (even) configuration, the discrepancy is bounded by 11.8 on all vertices at all times. Surprisingly, this constant improves to 10.9 for all rotor sequences (cf. Section 1.3), which do not update the rotor clockwise or counterclockwise. In Chapter 4 we show that there are configurations, which enforce a discrepancy of at least 7.2 on a single vertex. This time, there are configurations with higher discrepancy, namely 7.7, if the rotors do change clockwise or counterclockwise. This suggests that the single vertex discrepancy depends on the chosen rotor sequence.

In Chapter 5 we introduce a generalized “free” Propp machine. Instead of fixing a certain cyclic order and updating the rotors according to this, here we just require the rotors to change in such a way that the number of chips sent in each direction differs by at most a constant \( \Delta \). In this model, we can prove an upper bound of \( 26.9\Delta \) and a lower bound of \( 7.7\Delta \) for the discrepancy on a single vertex.

**1.2 Preliminaries and Notation**

First, we explain the technicality mentioned in Section 1.1. Without Assumption 2 below, also the results of [8, 9] are not valid. Since \( \mathbb{Z}^2 \) is bipartite, chips that start on even vertices (\( x \) with \( x_1 + x_2 \equiv 0 (\text{mod} 2) \)) never mix with those starting on odd positions (\( x \) with \( x_1 + x_2 \equiv 1 (\text{mod} 2) \)).

\(^1\)Note, that the definition of odd/even vertices will be changed after Assumption 2.
at once. However, this is not true, because chips of different parity may affect each other through the rotors. The odd chips may even reset the arrows and thus mess up the even chips and vice versa. Similar to the Arrow-forcing Theorem \[9\] from Section \[14\] one can cleverly arrange piles of off-parity chips to reorient the rotors and steer them away from random walk simulation. We therefore require the following.

**Assumption 1.** The starting configuration has chips only on one parity.

Without loss of generality, we consider only “even starting configurations”, where chips are only on even positions. Odd starting configurations can be handled in an analogous manner.

A random walk can be described nicely by its probability density \([26]\). We will draw on the following notation.

**Definition 1.** Let \(H(x, t)\) denote the probability that a chip from the origin arrives at location \(x\) at time \(t\) in a simple random walk.

Unfortunately, the equations for \(H(x, t)\) on the standard two-dimensional grid \(\mathbb{Z}^2\) (with arrows pointing up/down/left/right) shown in Figure 1(a) are not handy at all, because the two dimensions of \(\mathbb{Z}^2\) are not varying independently in a random walk. Since the quantities \(x_1 + x_2\) and \(x_1 - x_2\) are indeed varying independently,

![Figure 1](image-url)

**Figure 1:** Visualization of the applied coordinate transformation. Grid (b) can be seen as grid (a) rotated counterclockwise by 45° and scaled by \(\sqrt{2}\). The black and gray dots mark even and odd positions according, respectively. The red arrows indicate the neighborhood of the origin.
the coordinate transformation

\[
\begin{align*}
x'_1 & := x_1 + x_2 \\
x'_2 & := x_1 - x_2
\end{align*}
\] (1)

for all \((x_1, x_2) = x \in \mathbb{Z}^2\) results in a representation, where both dimensions vary independently at each step\(^2\). This is the reason for the following assumption.

**Assumption 2.** The neighbors of a vertex are in the directions

\[
\text{DIR} := \{ \nearrow, \searrow, \swarrow, \nwarrow \} := \{(1,1), (1,-1), (-1,1), (-1,-1)\}.
\]

Thus, chips move only on positions \(x \in \mathbb{Z}^2\) with \(x_1 + x_2 \equiv 0 \mod 2\). Via Equation (1) our results can immediately be translated into the standard two-dimensional grid model with neighbors \(\{1, \rightarrow, 1, \leftarrow\}\).

We need some notation to describe chips on the grid.

**Definition 2.** Let \(x \in \mathbb{Z}^2\) and \(t \in \mathbb{N}_0\), then \(x \sim t\) denotes \(x_1 \equiv x_2 \equiv t \mod 2\) and \(x \sim y\) denotes \(x_1 \equiv x_2 \equiv y_1 \equiv y_2 \mod 2\). A position \(x\) is called “even” or “odd” if \(x \sim 0\) or \(x \sim 1\), respectively. A configuration is called “even” or “odd” if all chips are at an even or all at an odd positions, respectively.

Therefore,

\[
H(x, t) = \begin{cases} 
4^{-t} \binom{t}{(t+x_1)/2} \binom{t}{(t+x_2)/2} & \text{if } x \sim t \text{ and } \|x\|_\infty \leq t \\
0 & \text{otherwise.}
\end{cases}
\]

We now fix a certain configuration and describe the state of both Propp machine and expected random walk.

**Definition 3.** For a fixed starting configuration, \(f(x, t)\) denotes the number of chips and \(\text{ARR}(x, t)\) denotes the direction of the arrow at position \(x\) after \(t\) steps of the Propp machine. \(E(x, t)\) denotes the expected number of chips at location \(x\) after running a random walk for \(t\) steps.

In the proofs, we also need the following mixed notation.

**Definition 4.** For a fixed starting configuration, \(E(x, t_1, t_2)\) denotes the expected number of chips at location \(x\) after \(t_1\) Propp and \(t_2 - t_1\) random steps.

\(^2\)Subscript \(k\) will always denote the \(k\)-th component of a vector.
1. Introduction

1.3 Rotor Sequence

To minimize the discrepancy between the number of chips sent left and right in the one-dimensional case, the arrow must switch back and forth after each chip sent. Two consecutive chips in the same direction would instantly increase the deviation of the Propp machine compared to the random walk. Hence, there is just one reasonable way to decide which arrow direction comes next, that is, to strictly alternate the arrows. This optimally equals out chips sent to the left and the right. In higher dimension this cannot be achieved for all dimensions at the same time.

On the two-dimensional grid $\mathbb{Z}^2$, Levine and Peres [29, 30] rotated the arrow clockwise by 90 degrees during each step. In our notation, they only considered the rotor sequence ($\rightarrow$, $\downarrow$, $\leftarrow$, $\uparrow$). Cooper and Spencer [10] were more general and allowed arbitrary cyclic orderings of the neighbors. Hence, we will also fix a certain permutation of the four directions and use this to define a cyclic function $\text{NEXT} : \text{DIR} \rightarrow \text{DIR}$ with $\text{DIR}$ as defined in Assumption 2 which gives the respective next arrow direction. With this, we can describe the resulting arrow after one Propp step as

$$\text{ARR}(x, t + 1) := \text{NEXT}^f(x, t)(\text{ARR}(x, t)).$$

Note that $\text{NEXT}^k$ can be calculated easily. So for example for the above mentioned rotor sequence ($\rightarrow$, $\downarrow$, $\leftarrow$, $\uparrow$), we have

$$\text{NEXT}^k(A) = \left(\begin{array}{c} (-1)\left[\frac{1}{2} - \frac{1}{4} + \frac{1}{4} \cdot (\frac{1}{4} + 1)\right], (-1)\left[\frac{1}{2} + \frac{1}{4} \cdot (\frac{1}{4} + 1)\right] \end{array}\right).$$

We now want to classify the $4! = 24$ permutations of the four arrows. Without loss of generality we fix the first arrow to $(\rightarrow_1) \equiv (\rightarrow)$ and thus only need to examine $3! = 6$ permutations. These differ on how equally they distribute the chips. How well a certain permutation equals out the chips sent in the four directions can be measured by colorings $\chi : \text{DIR} \rightarrow \{\pm 1\}$ of the arrows.

Without loss of generality we assume $\chi((\rightarrow_1)) = +1$ for all colorings. Then there are three balanced colorings, i.e., colorings with $|\chi^{-1}(+1)| = |\chi^{-1}(-1)| = 2$. These are $\chi_x\left(\begin{array}{c} x \end{array}\right) := x$, $\chi_y\left(\begin{array}{c} y \end{array}\right) := y$, and $\chi_{xy}\left(\begin{array}{c} x \end{array}\right) := xy$, as shown in Figure 2. $\chi_x$, $\chi_y$, and $\chi_{xy}$ measure how equal the arrows swap horizontally, vertically, and diagonally, respectively. A good permutation should minimize $\sum_{A \in \text{ARR}} |\chi(A) + \chi(\text{NEXT}(A))|$ for all three colorings $\chi$. 

![Figure 2: Colorings $\chi_x$, $\chi_y$, and $\chi_{xy}$ measure how equal the arrows swap horizontally, vertically, and diagonally, respectively. A good permutation should minimize $\sum_{A \in \text{ARR}} |\chi(A) + \chi(\text{NEXT}(A))|$ for all three colorings $\chi$.](image-url)
1. Related Work

**Diffusion-Limited Aggregation (DLA)**

DLA is a growth model on the infinite grid $\mathbb{Z}^d$, which was introduced in 1981 by Witten and Sander \cite{41, 42} to model aggregates of condensing metal vapor. It has since then attracted much interest\footnote{\cite{41} on this subject is the 9th most cited Physical Review Letter in the history of the journal.} among mathematicians and physicists. It simulates, for example, the coral growth, the path taken by a lightning, coalescing of dust or smoke particles, and the growth of some crystals \footnote{\cite{5}.}

In contrast to our Propp model, in DLA there is never more than one particle (chip) on a single point. The process starts with only the origin 0 being occupied. With each step, one particle diffuses in “from infinity” according to a random walk and attaches itself to the first boundary site visited. By the boundary of a cluster we mean all points adjacent, but not belonging to it. See Figure \ref{fig:example} for a typical example on $\mathbb{Z}^2$. Computer simulations suggest that after sufficiently many steps these clusters are highly ramified, of low density and their radius is of greater

<table>
<thead>
<tr>
<th>Arrow sequence</th>
<th>Permutation $\pi$</th>
<th>$\chi_x(\pi)$</th>
<th>$\chi_y(\pi)$</th>
<th>$\chi_{xy}(\pi)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\uparrow, \downarrow, \checkmark, \checkmark)</td>
<td>($\uparrow_1^1$, $\downarrow_1^1$, $\checkmark_1^1$, $\checkmark_1^1$)</td>
<td>(+, +, −, −)</td>
<td>(+, −, −, +)</td>
<td>(+, −, +, −)</td>
</tr>
<tr>
<td>(\uparrow, \downarrow, \checkmark, \downarrow)</td>
<td>($\uparrow_1^1$, $\downarrow_1^1$, $\checkmark_1^1$, $\downarrow_1^1$)</td>
<td>(+, +, −, +)</td>
<td>(+, +, −, +)</td>
<td>(+, +, −, +)</td>
</tr>
<tr>
<td>(\uparrow, \downarrow, \checkmark, \downarrow)</td>
<td>($\uparrow_1^1$, $\downarrow_1^1$, $\checkmark_1^1$, $\downarrow_1^1$)</td>
<td>(+, −, +, −)</td>
<td>(+, +, −, +)</td>
<td>(+, +, −, +)</td>
</tr>
<tr>
<td>(\uparrow, \checkmark, \checkmark, \downarrow)</td>
<td>($\uparrow_1^1$, $\checkmark_1^1$, $\checkmark_1^1$, $\downarrow_1^1$)</td>
<td>(+, −, −, −)</td>
<td>(+, +, +, −)</td>
<td>(+, +, −, +)</td>
</tr>
<tr>
<td>(\uparrow, \checkmark, \checkmark, \downarrow)</td>
<td>($\uparrow_1^1$, $\checkmark_1^1$, $\checkmark_1^1$, $\downarrow_1^1$)</td>
<td>(+, +, +, +)</td>
<td>(+, −, −, +)</td>
<td>(+, +, −, +)</td>
</tr>
</tbody>
</table>

Table 1: All six permutations and their respective ±1 sequences for each coloring.

For the sake of readability we use “+” and “−” as short form of $+1$ and $−1$. Table 1 shows that each permutation only strictly alternates on exactly one coloring and alternates in groups of two on the other two colorings. Note that the two probably most natural permutations, where the arrow goes in turns, i.e. (\uparrow, \downarrow, \checkmark, \checkmark) and (\uparrow, \downarrow, \checkmark, \downarrow), are the two most unbalanced in terms of vertical and horizontal balance.

Beside fixing a certain permutation, there are other models how to define NEXT. These are discussed in Chapter 5.

1.4 Related Work
1. Introduction

Related Work

order of magnitude than the $d$-th root of the number of particles \cite{15}. Martin T. Barlow \cite{1} conjectured that the resulting cluster has a fractal dimension exceeding 1, but rigorous results are hard to establish in this model.

Internal Diffusion-Limited Aggregation (IDLA)

Diaconis and Fulton \cite{13} introduced Internal DLA in 1991 as a variant of classical DLA. There, the random walk begins at the origin and the particle occupies the first empty site it reaches. This yields a central growing blob. Since the next particle is more likely to fill an unoccupied site close to the origin than one further away, one intuitively expects this blob to be a sphere. Figure 4 shows three IDLA clusters with different numbers of particles accumulated. Lawler, Bramson, and Griffeath \cite{27} showed in 1992 that the limiting shape after $n$ steps is indeed a Euclidean ball as $n \to \infty$. In a subsequent paper, Lawler \cite{25} proved that with high probability the fluctuations around this limiting shape are bounded by $O(n^{1/3})$. Moore and Machta \cite{32} observed experimentally that these error terms were even smaller, namely of logarithmic size. Soon after, Blachère \cite{4} claimed to have shown that the fluctuations are at most logarithmic, but there is an error in his main theorem. Consequently, it is still unknown how to bridge this gap between theory and observation.

![Figure 3: A typical example of two-dimensional diffusion limited aggregation. – Thanks to Paul Bourke from the Centre for Astrophysics and Supercomputing, Swinburne University for providing the image 5.](image-url)
1. Introduction

Related Work

Figure 4: Internal DLA clusters with different number of particles. – Thanks to Jon Machta from the Department of Physics at the University of Massachusetts Amherst and Cristopher Moore from the Santa Fe Institute in New Mexico for providing the images [32].

Aggregating Propp Machines

The Propp machine can also be used to define a deterministic aggregation model analogous to IDLA. Instead of moving all chips in each step, one then repeatedly adds a single chip at the origin and performs a rotor router walk until the chip reaches an unoccupied position and occupies it\(^4\). Levine and Peres [29, 30] proved that the shape of this model in \(\mathbb{Z}^d\) converges to a Euclidean ball in \(\mathbb{R}^d\). More precisely, [28] shows that after \(n\) chips added, this “Propp circle” contains a disk whose radius grows as \(n^{1/4}\). See Figure 5 for an example. Kleber [21] adds interesting experimental results for \(\mathbb{Z}^2\), if the rotor is changing clockwise.

After three million chips, the occupied site furthest from the origin is at distance \(\sqrt{956609} \approx 978.06\), while the unoccupied site nearest the origin is at distance \(\sqrt{953461} \approx 976.45\), a difference of about 1.61. Hence, the occupied sites almost form a perfect circle.

We reran Klebers experiment and discovered that for the aggregating Propp machine, the chosen rotor sequence matters as well. The difference of 1.61 reported by Kleber [21] only occurs for clockwise and counterclockwise sequences. For other rotor sequences we either have 0.92 or 1.85, depending on whether the first change of the rotor is by 90° or 180°. This supports our conjecture that also for our Propp model introduced in Section 1.1 we expect different discrepancies depending on the rotor sequence.

\(^4\)At http://www.math.wisc.edu/~propp/rotor-router-1.0/ one can find a nice Java applet by Hal Canary and Francis Wrong, demonstrating this model.
1. Introduction

Related Work

Figure 5: The rotor-router blob (also known as “Propp Circle”) after 750,000 chips. Every site’s rotor begins pointing East and is rotated counterclockwise. The color indicates in which direction the rotor is pointing. – Thanks to Lionel Levine from UC Berkeley and Ed Pegg Jr. from Wolfram Research, Inc for providing the image [33].

Chip-firing Games

A chip-firing game can take place on an arbitrary (directed) graph [2, 3]. Each node contains a pile of chips. A move consists of selecting a node with at least as many chips as its (out)degree, and sending one chip along each (outgoing) edge to its neighbors. In full generality, chips can also possibly disappear permanently from the system. This mechanism was first described, under the name “probabilistic abacus” by Arthur Engel in 1975 in a math eduction journal [16, 17]. There are many results on aspects such as its entropy, reachability, and periodicity.

The “abelian sandpile model” is a special chip-firing game, usually discussed on $\mathbb{Z}^2$ [12]. There, each node holds a pile of sand. If any pile has five or more grains of sand, it collapses with four grains sliding off of it and getting dumped on the node’s four neighbors. This then, in turn, makes some neighboring piles unstable and causes further topplings until each pile has size of at most four.
Load-Balancing

In parallel computing, load-balancing a distributed network is an important task. The processors are modeled as the vertices of an undirected connected graph and links between them as edges. Initially, each processor has a collection of unit-size tasks. The object is to balance the number of tasks at each processor by transmitting tasks along edges according to some local scheme. Dynamic load balancing is crucial for the efficiency of highly parallel systems when solving non-uniform problems with unpredictable load estimates [40].

Diffusion is one well-established algorithm for load balancing. If a processor has more tasks than one of its neighbors, it sends some of its task to the neighbor. The number sent is proportional to the differential between the two processors and some constant depending on the connecting edge. A standard choice for these constants is uniform diffusion, in which each processor simply averages the loads of its neighbors at each step [36].

There are some thousand research papers on dynamic load balancing. Many model load-balancing algorithms as a Markov chain, which converges to the uniform load vector. Thereby, they allow sending fractional tasks and ignore the rounding to whole tasks at each local balancing step. This deviation can be large depending on the specific network topology [18, 36, 39].
Chapter 2

The Basic Method

This chapter gives the basic formulas to compare Propp machine and expected random walk based on the number of chips on a single vertex.

2.1 Characterizing the Discrepancy

In order to avoid discussing all equations in the expected sense and thereby to simplify the presentation, one can treat the expected random walk as a “linear machine” \[10\]. Here, in each time step a pile of \( k \) chips is split evenly, with \( k/4 \) chips going to each neighbor. The (possibly non-integral) number of chips at position \( x \) at time \( t \) is exactly the expected number of chips \( E(x, t) \) in the random walk model.

Using the notation of Definition \[3\] we are interested in bounding the discrepancies \( f(x, t) - E(x, t) \) for all vertices \( x \) and all times \( t \). Via translation on the infinite grid, it suffices to prove the result for the vertex \( x = 0 \). Clearly,

\[
E(0, 0, t) = E(0, t), \\
E(0, t, t) = f(0, t),
\]

so that

\[
f(0, t) - E(0, t) = \sum_{s=0}^{t-1} (E(0, s + 1, t) - E(0, s, t)). \tag{2}
\]
Each summand of Equation (2) characterizes the difference whether a chip first makes a Propp step and then goes \( t - s - 1 \) linear steps or it goes \( t - s \) linear steps in the first place. Note that positions, where the number of chips are multiples of four, do not contribute to the total sum at all. By \( \text{REM}_j^{(s)} \) we denote the set of locations that are occupied by \( k \) chips with \( k \equiv j \pmod{4} \) at time \( s \), i.e., for \( 0 \leq j \leq 3 \), we set

\[
\text{REM}_j^{(s)} := \{ x \mid f(x, s) \equiv j \pmod{4} \}.
\]

We therefore get

\[
E(0, s + 1, t) - E(0, s, t) = \sum_{y \in \text{REM}_1^{(s)}} \left( H(y + \text{ARR}(y, s), t) - H(y, t) \right) \\
+ \sum_{y \in \text{REM}_2^{(s)}} \left( H(y + \text{ARR}(y, s), t - s - 1) + H(y + \text{NEXT}(\text{ARR}(y, s)), t - s - 1) - 2H(y, t - s) \right) \\
+ \sum_{y \in \text{REM}_3^{(s)}} \left( H(y + \text{ARR}(y, s), t - s - 1) + H(y + \text{NEXT}(\text{ARR}(y, s)), t - s - 1) + H(y + \text{NEXT}^2(\text{ARR}(y, s)), t - s - 1) - 3H(y, t - s) \right).
\]

To sum this up over all times \( s \), we have to look at single chips. Let \( s_i(y) \) denote the time that \( y \) is occupied by its \( i \)-th chip:

**Definition 5.** For all \( i \in \mathbb{N}_0 \), \( s_i(y) := \min \left\{ u \geq 0 \mid i < \sum_{t=0}^{u} f(y, t) \right\} \).

Now we can combine Equations (2) and (3) to

\[
f(0, t) - E(0, t) = \sum_{y \in \mathbb{Z}^2} \sum_{i \geq 0} \left( H(y + A_i(y), t - s_i(y) - 1) - H(y, t - s_i(y)) \right)
\]  

with \( A^i(y) := \text{ARR}(y, s_i(y)) \). Note that \( \text{ARR}(y, s_i(y)) = \text{NEXT}^i(\text{ARR}(y, 0)) \) and that this can be calculated easily.

---

1 Since four chips at a common location are distributed equally, it actually suffices to count only remainders modulo 4, i.e., \( f(y, t) \pmod{4} \), for the regular Propp machine. However, the free Propp machine of Chapter 5 has to regard multiples of four. Hence, to get a common notation, we count each chip.
Combining Equations (4) and (5), we now obtain our Lemma 1, which will be

**Lemma 1.** For all times $t \in \mathbb{N}_0$,

$$f(0, t) - E(0, t) = \sum_{y \in \mathbb{Z}^2} \left( \sum_{i \geq 0} \frac{-A_1^{(i)}(y)y_1}{t - s_i(y)} H(y, t - s_i(y)) + \sum_{i \geq 0} \frac{-A_2^{(i)}(y)y_2}{t - s_i(y)} H(y, t - s_i(y)) + \sum_{i \geq 0} \frac{A_1^{(i)}(y)A_2^{(i)}(y)y_1y_2}{(t - s_i(y))^2} H(y, t - s_i(y)) \right).$$

with $A^{(i)}(y) := \text{ARR}(y, s_i(y))$ and $s_i(y)$ as in Definition 5.

The last equality holds due to $\left(\frac{t-1}{(t-1+y_1+A_1)/2}\right) / \left(\frac{t}{(t+y_1)/2}\right) = \frac{1}{2} \left(1 - A_i^2\right)$ for $A = \pm 1$ and $y_1, y_2 \leq t$.

Combining Equations (4) and (5), we now obtain our Lemma 1 which will be crucial in the remainder:

**Lemma 1.** For all times $t \in \mathbb{N}_0$,

$$H(y + A, t - 1) - H(y, t)$$

$$= 4^{1-t} \left( (t - 1 + y_1 + A_1)/2 \right) \left( (t - 1 + y_2 + A_2)/2 \right) - 4^{-t} \left( (t + y_1)/2 \right) \left( (t + y_2)/2 \right)$$

$$= \left( (t - 1)^2 4^{1-t} \left( \frac{t}{2} \right) \right) \left( (t - 1)^2 4^{1-t} \left( \frac{t}{2} \right) \right)$$

$$= \left( (t - 1)^2 4^{1-t} \left( \frac{t}{2} \right) \right) \left( (t - 1)^2 4^{1-t} \left( \frac{t}{2} \right) \right)$$

$$= \left( (t - 1)^2 4^{1-t} \left( \frac{t}{2} \right) \right) \left( (t - 1)^2 4^{1-t} \left( \frac{t}{2} \right) \right)$$

$$= \left( (t - 1)^2 4^{1-t} \left( \frac{t}{2} \right) \right) \left( (t - 1)^2 4^{1-t} \left( \frac{t}{2} \right) \right)$$

$$= \left( (t - 1)^2 4^{1-t} \left( \frac{t}{2} \right) \right) \left( (t - 1)^2 4^{1-t} \left( \frac{t}{2} \right) \right)$$

The last equality holds due to $\left(\frac{t-1}{(t-1+y_1+A_1)/2}\right) / \left(\frac{t}{(t+y_1)/2}\right) = \frac{1}{2} \left(1 - A_i^2\right)$ for $A = \pm 1$ and $y_1, y_2 \leq t$.
2.2 Unimodality

To calculate the discrepancy between the Propp and the linear machine, we have to bound the alternating sums of Lemma 1. To achieve this, we will use the following property of the involved functions:

**Definition 6.** Let $X \subseteq \mathbb{R}$. We call a mapping $f : X \to \mathbb{R}$ unimodal, if there is an $m \in X$ such that $f$ is monotonically increasing in $\{ x \in X \mid x \leq m \}$ and $f$ is monotonically decreasing in $\{ x \in X \mid x \geq m \}$.

Unimodal functions are popular in optimization and probability theory. The probability $H(x, t)$ that a chip from the origin arrives at location $x$ at time $t$ in a simple random walk is unimodal in $t$. This is depicted in Figure 6. The specific shape can be explained easily. For $t < \|x\|_\infty$ the probability that a chip from the origin arrives at $x$ is zero. With more time elapsed, the probability raises due to the increasing number of paths of length $t$ to $x$, but since the number of reachable positions rises quadratically, the probability decreases again after a certain threshold. That $H(x, t)$ is indeed unimodal in $t$ can be proven with Sturm’s theorem \[38\]. For Lemma 7 we will require unimodality of $H(x, t)/t$ and $H(x, t)/t^2$ in $t$. This is less intuitive than the unimodality of $H(x, t)$ and will be proven in Lemmas 4 and 5.

Our interest in unimodality of the functions $H(x, t)/t$ and $H(x, t)/t^2$ from Lemma 4 is due to the following lemma.

![Figure 6: Plot of $H(^6_1, t)$ as an example of a unimodal function.](image-url)
Lemma 2. Let $f : X \to \mathbb{R}$ be non-negative and unimodal. Let $t_0, \ldots, t_n \in X$ such that $t_0 \leq \ldots \leq t_n$. Then

$$\left| \sum_{i=0}^{n} (-1)^i f(t_i) \right| \leq \max_{x \in X} f(x).$$

Proof. Let $\zeta$ be even with $f(t_\zeta) = \max_{i \text{ even}} f(t_i)$ and $\xi$ be odd with $f(t_\xi) = \max_{i \text{ odd}} f(t_i)$. By these definitions we ensure that $f|_{\leq \zeta}$ and $f|_{\leq \xi}$ are monotonically increasing and $f|_{\geq \zeta}$ and $f|_{\geq \xi}$ are monotonically decreasing. Therefore

$$\sum_{i=0}^{n} (-1)^i f(t_i) = \sum_{i=0}^{\zeta} (-1)^i f(t_i) + \sum_{i=\zeta+1}^{\xi} (-1)^i f(t_i) + \sum_{i=\xi+1}^{n} (-1)^i f(t_i) \leq f(t_\zeta) \leq \max_{x \in X} f(x)$$

and

$$\sum_{i=0}^{n} (-1)^i f(t_i) = \sum_{i=0}^{\zeta} (-1)^i f(t_i) + \sum_{i=\zeta+1}^{\xi} (-1)^i f(t_i) + \sum_{i=\xi+1}^{n} (-1)^i f(t_i) \geq -f(t_\xi) \geq -\max_{x \in X} f(x)$$

give

$$-\max_{x \in X} f(x) \leq \sum_{i=0}^{n} (-1)^i f(t_i) \leq \max_{x \in X} f(x).$$

If the sums of Lemma 1 are indeed alternating and if the involved functions are unimodal, we could now apply Lemma 2 to calculate the desired discrepancy bound. But first, we need to prove that $H(x, t)/t$ and $H(x, t)/t^2$ are indeed unimodal in $t$. For these proofs we need the following rule, which can be found in [1, 22, 23]:

Theorem 3 (Descartes’ Rule of Signs). The number of positive roots (counting multiplicities) of a non-zero polynomial with real coefficients is either equal to its number of coefficient sign variations or else is less than this number by an even integer.
With this, we are now well equipped to prove unimodality of $H(x, t)/t$ and $H(x, t)/t^2$. For the calculations in Sections 3.2 and 3.3, we also already derive the exact times, when these discrete functions are maximal.

**Lemma 4.** $H(x, t)/t$ is a unimodal function in $t$ with its global maximum at $t_{\text{max}}(x) \in \left[ \frac{x_1^2 + x_2^2}{4} - 2, \frac{x_1^2 + x_2^2}{4} + \frac{1}{2} \right]$.

**Proof.** Without loss of generality $x_1 + x_2 > 4$ and $0 \leq x_1 \leq x_2$. By definition $H(x, t)/t = 0$ for $t < x_2^2$. We want to show $H(x, t)/t$ has only one maximum in $t \in [x_2, \infty)$. Consequently, we compute the discrete derivative of $H(x, t)/t$ via

$$
\frac{H(x, u - 2) - H(x, u)}{u + 2} = \frac{4^{-u} \left( 4u^3 - (x_1^2 + x_2^2 + 5)u^2 + 2u + x_1^2x_2^2 \right) (u - 3)! (u - 2)!}{(u + x_1^2)! (u - x_2^2)! (u + x_2^2)! (u - x_1^2)!}.
$$

Since all but one factors are always positive, it suffices to show that

$$
p(u) := 4u^3 - (x_1^2 + x_2^2 + 5)u^2 + 2u + x_1^2x_2^2
$$

has not more than one real root greater than $x_2$. This can be done with Descartes’ Sign Rule described in Theorem 3. Since we assumed $x_1$ and $x_2$ to be nonnegative, $p(u)$ has two sign changes. Descartes’ Rule now says that $p(u)$ has exactly zero or two positive real roots. With

$$
p(u - x_2) = 4u^3 - (x_1^2 + x_2^2 + 12x_2 + 5)u^2
+ (2x_2^3 + 12x_2^2 + 2x_1^2x_2 + 10x_2 + 2)u - (x_2^4 + 4x_2^3 + 5x_2^2 + 2x_2)
$$

Descartes furthermore testifies that $p$ has exactly one or three real roots in $[x_2, \infty)$. Both implications together yield the desired result that there is only one real root in $[x_2, \infty)$.

To find the global maximum $t_{\text{max}}(x)$ of $H(x, t)/t$, the cubic formula gives us the exact position of the only relevant root $u_0$ of $p(u)$:

$$
u_0 = \frac{1}{12} \left( x_1^2 + x_2^2 + 5 + \frac{x_1^4 + x_2^4 + 2x_1^2x_2^2 + 10x_1^2 + 10x_2^2 + 1}{P} \right)
$$

17
with
\[ P := \sqrt[3]{Q + 12\sqrt{3}R} \]
\[ Q := x_1^6 + x_2^6 + 3x_1^2x_2^4 + 3x_1^4x_2^2 + 15x_1^4 + 15x_2^4 - 186x_1^2x_2^2 + 39x_1^2 + 39x_2^2 - 55 \]
\[ R := -x_1^8x_2^2 - x_1^6x_2^4 - 3x_1^4x_2^6 - 3x_2^6x_1^4 + 78x_2^4x_1^2 - 15x_1^6x_2^2 - 15x_1^2x_2^6 \]
\[ - 39x_1^4x_2^2 - 39x_1^2x_2^4 - x_1^4 - x_2^4 + 53x_1^2x_2^2 - 10x_1^2 - 10x_2^2 + 7. \]

Note that the derived \( u_0 \) is fractional, though we actually require \( t_{\max}(x) \in \mathbb{Z} \) and \( x \sim t_{\max}(x) \). An exact formula for the integer \( t_{\max}(x) \) can be obtained by observing for all \( x \)
\[ p \left( \frac{x_1^2 + x_2^2}{4} \right) = \frac{1}{16} \left( 6x_1^2x_2^2 + 8x_1^2 + 8x_2^2 - 5x_1^4 - 5x_2^4 \right) < 0 \]
and
\[ p \left( \frac{x_1^2 + x_2^2 + 5}{4} \right) = x_1^2x_2^2 + \frac{x_1^2 + x_2^2 + 5}{2} > 0, \]
which implies \( \frac{x_1^2 + x_2^2}{4} - 2 \leq t_{\max}(x) \leq \frac{x_1^2 + x_2^2 - 3}{4} \) for all \( x \) with \( x_1 + x_2 > 4 \).

Adding the boundary cases and the condition \( x \sim t_{\max}(x) \) this leads uniquely to
\[
 t_{\max}(x) = \begin{cases} 
 \|x\|_{\infty} & \text{if } \|x\|_1 \leq 4 \\
 \frac{x_1^2 + x_2^2}{4} - 2 & \text{if } x_1 \equiv x_2 \equiv 0 \pmod{2} \text{ and } x_1 + x_2 \equiv 0 \pmod{4} \\
 \frac{x_1^2 + x_2^2}{4} - 1 & \text{if } x_1 \equiv x_2 \equiv 0 \pmod{2} \text{ and } x_1 + x_2 \equiv 2 \pmod{4} \\
 \frac{x_1^2 + x_2^2}{4} - \frac{3}{2} & \text{if } x_1 \equiv x_2 \equiv 1 \pmod{2} 
\end{cases} \tag{6}
\]

In Section 3.2 we will need the fact that \( t_{\max}(x) \) can also be bounded multiplicatively by
\[ \frac{3}{16}(x_1^2 + x_2^2) \leq t_{\max}(x) \leq \frac{1}{2}(x_1^2 + x_2^2) \text{ for all } x. \tag{7} \]

\[ \square \]

**Lemma 5.** \( H(x, t)/t^2 \) is a unimodal function in \( t \) with its global maximum at \( t'_{\max}(x) \in \left[ \frac{x_1^2 + x_2^2}{6} - \frac{4}{3}, \frac{x_1^2 + x_2^2}{6} + \frac{4}{3} \right] \).

**Proof.** The proof proceeds completely analogous to the proof of Lemma 4. Without loss of generality \( x_1 + x_2 > 6 \) and \( 0 \leq x_1 \leq x_2 \). By definition \( H(x, t)/t^2 = 0 \)
for \( t < x_2 \). We want to show that \( H(x, t)/t^2 \) has only one maxima in \( t \in [x_2, \infty) \).

The discrete derivative of \( H(x, t)/t^2 \) is

\[
\frac{H(x, u - 2) - H(x, u)}{(u - 2)^2} = \frac{4^{-u}(6u^3 - (x_1^2 + x_2^2 + 13)u^2 + 12u + x_1^2x_2^2 - 4)((u - 3)!)}{(u - x_1/2)!((u - x_2/2)!((u - x_2)!).
\]

Since again all but one factors are always positive, it suffices to show that

\[
p(u) := 6u^3 - (x_1^2 + x_2^2 + 13)u^2 + 12u + (x_1^2x_2^2 - 4)
\]

has not more than one real root greater than \( x_2 \). Due to \( x_1 + x_2 > 6 \), \( p(u) \) has two sign changes and therewith zero or two positive real roots.

\[
p(u - x_2) = 6u^3 - (x_1^2 + x_2^2 + 18x_2 + 13)u^2 + (2x_1^2x_2 + 2x_2^2 + 18x_2^2 + 26x_2 + 12)u
\]

\[
- (x_1^2 + 6x_2^2 + 13x_2^2 + 12x_2 + 4)
\]

has three sign changes, i.e., \( p \) has exactly one real root in \([x_2, \infty)\).

To find the global maximum \( t'_{\text{max}}(x) \) of \( H(x, t)/t^2 \), the cubic formula gives us the exact position of the only relevant root \( u_0 \) of \( p(u) \):

\[
u_0 = \frac{1}{18} \left( x_1^2 + x_2^2 + 13 + \frac{x_1^4 + x_2^4 + 2x_1^2x_2^2 + 26x_1^2 + 26x_2^2 - 47}{P} \right)
\]

with

\[
P := 3\sqrt{Q + 18\sqrt{3R}}
\]

\[
Q := x_1^6 + x_2^6 + 3x_1^4x_2^2 + 3x_1^2x_2^4 + 39x_1^4 + 39x_2^4 + 183x_1^2 + 183x_2^2 - 408x_1^2x_2^2 - 71
\]

\[
R := -x_1^8x_2^2 - x_1^6x_2^4 - 3x_1^6x_2^4 - 3x_4^4x_2^4 - 39x_1^6x_2^2 - 39x_1^6x_2^2 + 165x_1^4x_4^4
\]

\[
+ 4x_4^6 + 4x_6^6 - 171x_1^4x_2^4 - 171x_1^3x_2^4 + 120x_1^4 + 120x_2^4 + 311x_1^2x_2^2
\]

\[
- 204x_1^2 - 204x_2^2 + 112.
\]

To get an exact formula for \( t'_{\text{max}}(x) \), it is again easy to see that for all \( x \)

\[
p \left( \frac{x_1^2 + x_2^2 + 3}{6} \right) = \frac{1}{18} \left( 6x_1^2 + 6x_2^2 + 8x_1^2x_2^2 - 5x_1^4 - 5x_2^4 - 9 \right) < 0
\]
2. The Basic Method

and

\[ p \left( \frac{x_1^2 + x_2^2 + 13}{6} \right) = 2x_1^2 + 2x_2^2 + x_1x_2^2 + 22 > 0, \]

which implies

\[ \frac{x_1^2 + x_2^2 + 3}{6} - 2 \leq t'_\text{max}(x) \leq \frac{x_1^2 + x_2^2 + 1}{6} \]

for all \( x \) with \( x_1 + x_2 > 6 \).

Adding the boundary cases and the condition \( t'_\text{max}(x) \equiv x_1 \equiv x_2 \pmod{2} \) this leads uniquely to

\[
 t'_\text{max}(x) = \begin{cases} 
 \|x\|_\infty & \text{if } \|x\|_1 \leq 6 \\
 \frac{x_1^2 + x_2^2}{6} - \frac{4}{3} & \text{if } x_1 \not\equiv 0 \not\equiv x_2 \pmod{3} \\
 \frac{x_1^2 + x_2^2}{6} - \frac{2}{3} & \text{if } x_1 \not\equiv 0 \equiv x_2 \pmod{3} \text{ or } x_1 \equiv 0 \not\equiv x_2 \pmod{3} \\
 \frac{x_1^2 + x_2^2}{6} & \text{if } x_1 \equiv 0 \equiv x_2 \pmod{3} 
\end{cases}
\]  

(8)

Note for Section 3.2 that \( t'_\text{max}(x) \) can also be bounded multiplicatively by

\[ \frac{1}{5}(x_1^2 + x_2^2) \leq t'_\text{max}(x) \leq \frac{1}{7}(x_1^2 + x_2^2) \text{ for all } x. \]  

(9)
Building on Lemma 1, we utilize Lemma 2, 4, and 5 of the previous chapter to prove the following upper bound on the single vertex discrepancy.

**Theorem 6.** Independent of the initial even configuration, the time $t$, and the location $x$, we have for clockwise $(\uparrow, \downarrow, \leftarrow, \rightarrow)$ and counterclockwise $(\downarrow, \uparrow, \rightarrow, \leftarrow)$ rotor sequences

$$|f(x, t) - E(x, t)| < 11.8,$$

and for all other rotor sequences, i.e., for $(\downarrow, \leftarrow, \rightarrow, \uparrow)$, $(\leftarrow, \rightarrow, \uparrow, \downarrow)$, $(\rightarrow, \uparrow, \downarrow, \leftarrow)$, and $(\rightarrow, \downarrow, \uparrow, \leftarrow)$,

$$|f(x, t) - E(x, t)| < 10.9.$$

### 3.1 Dependence on the Chosen Rotor Sequence

It remains to decompose the sums of Lemma 1 in strictly alternating sums of unimodal functions. Here a crucial difference to the one-dimensional case emerges: this decomposition, and therewith the upper bound, depends on the chosen rotor sequence (cf. Section 1.3).

We fix a $y \in \mathbb{Z}^2$ and set $A^{(i)}(y) := A^{(i)}(y) = \text{ARR}(y, s_i(y))$ and $t^{(i)} := t - s_i(y)$. For all three inner sums of Lemma 1 we now have to distinguish how the arrow alternates. Let us first look at the sum $\sum_{i \geq 0} \frac{-A^{(i)}(y)}{t^{(i)}} H(y, t^{(i)})$. For both rotor sequences $\pi \in \{(\uparrow, \downarrow, \leftarrow, \rightarrow), (\rightarrow, \downarrow, \uparrow, \leftarrow)\}$ we have $\chi_x(\pi) = (+, -, +, -)$ and $A_1^{(i)}$ strictly alternating. Hence, there is an $\alpha \in \{0, 1\}$ that only depends on
ARR(y, 0) such that
\[
\sum_{i \geq 0} \frac{-A_1^{(i)} y_1}{t^{(i)}} H(y, t^{(i)}) = (-1)^a \sum_{i \geq 0} (-1)^i y_1 H(y, t^{(i)})
\]
\[
= \left| \sum_{i \geq 0} (-1)^i \frac{y_1 H(y, t^{(i)})}{t^{(i)}} \right|.
\]
Since we have proven unimodality of \( H(x, t) / t \) in Lemma 4, we can now apply Lemma 2 and obtain
\[
\left| \sum_{i \geq 0} \frac{-A_1^{(i)} y_1}{t^{(i)}} H(y, t^{(i)}) \right| \leq \max_i \left| \frac{y_1 H(y, t)}{t} \right|.
\]
If the rotor sequence does not strictly alternate in the first dimension, we have to be more careful. For the remaining permutations we have either \( \chi_x(\pi) = (+, +, -, -) \) or \( \chi_x(\pi) = (+, -, -, +) \). This means that we can divide the sum in two alternating sums, which yields
\[
\left| \sum_{i \geq 0} \frac{-A_1^{(i)} y_1}{t^{(i)}} H(y, t^{(i)}) \right|
\leq \left| \sum_{i \geq 0} \frac{-A_1^{(i)} y_1}{t^{(i)}} H(y, t^{(i)}) \right|_{i \text{ odd}} + \left| \sum_{i \geq 0} \frac{-A_1^{(i)} y_1}{t^{(i)}} H(y, t^{(i)}) \right|_{i \text{ even}}
\]
\[
= \left| \sum_{i \geq 0} (-1)^i \frac{y_1 H(y, t^{(i)})}{t^{(i)}} \right|_{i \text{ odd}} + \left| \sum_{i \geq 0} (-1)^i \frac{y_1 H(y, t^{(i)})}{t^{(i)}} \right|_{i \text{ even}}
\]
\[
\leq 2 \max_i \left| \frac{y_1 H(y, t)}{t} \right|.
\]
For the second sum of Lemma 4 one can analogously prove
\[
\left| \sum_{i \geq 0} \frac{-A_2^{(i)} y_2}{t^{(i)}} H(y, t^{(i)}) \right| \leq \max_i \left| \frac{y_2 H(y, t)}{t} \right|
\]
for the permutations (\( \nearrow, \searrow, \swarrow, \nwarrow \)) and (\( \nearrow, \searrow, \swarrow, \nwarrow \)), which satisfy \( \chi_y(\pi) = (+, -, +, -) \). Also
\[
\left| \sum_{i \geq 0} \frac{-A_2^{(i)} y_2}{t^{(i)}} H(y, t^{(i)}) \right| \leq 2 \max_i \left| \frac{y_2 H(y, t)}{t} \right|
\]
3. Upper Bound Dependence on the Chosen Rotor Sequence

for the other four permutations with \( \chi_y(\pi) = (+, +, -, -) \) or \( \chi_y(\pi) = (+, -, -, +) \).

The third sum of Lemma 1 can also be bounded by

\[
\left| \sum_{i \geq 0} A_1^{(i)} A_2^{(i)} y_1 y_2 \frac{H(y, t^{(i)})}{t^{(i)}} \right| \leq \max_t \left| \frac{y_1 y_2 H(y, t)}{t^2} \right|
\]

for the permutations \( (\swarrow, \searrow, \nearrow, \nwarrow) \) and \( (\searrow, \swarrow, \nearrow, \nwarrow) \) with \( \chi_{xy}(\pi) = (+, -, +, -) \) and by

\[
\left| \sum_{i \geq 0} A_1^{(i)} A_2^{(i)} y_1 y_2 \frac{H(y, t^{(i)})}{t^{(i)}} \right| \leq 2 \max_t \left| \frac{y_1 y_2 H(y, t)}{t^2} \right|
\]

for the four permutations with \( \chi_{xy}(\pi) = (+, +, -, -) \) or \( \chi_{xy}(\pi) = (+, -, -, +) \). Due to symmetry in the two dimensions, we also have

\[
\max_t \left| \frac{y_1 H(y, t)}{t} \right| = \max_t \left| \frac{y_2 H(y, t)}{t} \right|
\]

Putting all this together, we obtain

**Lemma 7.** Independent of the initial even configuration and the time \( t \), we have for clockwise \( (\swarrow, \searrow, \nearrow, \nwarrow) \) and counterclockwise \( (\searrow, \swarrow, \nearrow, \nwarrow) \) rotor sequences

\[
|f(0, t) - E(0, t)| \leq \sum_{y \in \mathbb{Z}^2} \left( 4 \max_t \left| \frac{y_1 H(y, t)}{t} \right| + \max_t \left| \frac{y_1 y_2 H(y, t)}{t^2} \right| \right)
\]

and for all other rotor sequences, i.e., for \( (\swarrow, \searrow, \nwarrow, \nearrow) \), \( (\nearrow, \swarrow, \searrow, \nwarrow) \), \( (\searrow, \nwarrow, \swarrow, \nearrow) \), and \( (\nwarrow, \swarrow, \nearrow, \searrow) \),

\[
|f(0, t) - E(0, t)| \leq \sum_{y \in \mathbb{Z}^2} \left( 3 \max_t \left| \frac{y_1 H(y, t)}{t} \right| + 2 \max_t \left| \frac{y_1 y_2 H(y, t)}{t^2} \right| \right).
\]
3. Upper Bound

3.2 Convergence of the Series

To obtain the desired numerical bounds via Lemma 7, we want to calculate \( \sum_{y} \max_{t} \left| \frac{y_{1}H(y, t)}{t} \right| \) and \( \sum_{y} \max_{t} \left| \frac{y_{1}y_{2}H(y, t)}{t^{2}} \right| \). This depends on the convergence of both series.

The summands can be bounded as follows. By Lemma 14 and Equation (7),

\[
\max_{t} \left| \frac{y_{1}H(y, t)}{t} \right| = \frac{|y_{1}|}{t_{\max}(y)} H(y, t_{\max}(y)) \\
\leq \frac{|y_{1}|}{t_{\max}(y)} 4^{-t_{\max}(y)} \left( \frac{t_{\max}(y)}{t_{\max}(y) + y_{1}/2} \right) \left( \frac{t_{\max}(y)}{t_{\max}(y) + y_{2}/2} \right) \\
\leq \frac{|y_{1}|}{t_{\max}(y)} \left( 2^{-t_{\max}(y)} \left( \frac{t_{\max}(y)}{t_{\max}(y)/2} \right) \right)^{2} \\
\leq \frac{256 |y_{1}|}{9 (y_{1}^{2} + y_{2}^{2})^{2}}, \tag{10}
\]

and analogously by Lemma 14 and Equation (9),

\[
\max_{t} \left| \frac{y_{1}y_{2}H(y, t)}{t^{2}} \right| = \frac{|y_{1}y_{2}|}{t'_{\max}(y)} H(y, t'_{\max}(y)) \\
\leq \frac{|y_{1}y_{2}|}{t'_{\max}(y)} 4^{-t'_{\max}(y)} \left( \frac{t'_{\max}(y)}{t'_{\max}(y) + y_{1}/2} \right) \left( \frac{t'_{\max}(y)}{t'_{\max}(y) + y_{2}/2} \right) \\
\leq \frac{|y_{1}y_{2}|}{t'_{\max}(y)} \left( 2^{-t'_{\max}(y)} \left( \frac{t'_{\max}(y)}{t'_{\max}(y)/2} \right) \right)^{2} \\
\leq \frac{83 |y_{1}y_{2}|}{(y_{1}^{2} + y_{2}^{2})^{3}} \leq \frac{256}{(y_{1}^{2} + y_{2}^{2})^{2}}, \tag{11}
\]

The last inequality is due to \( \frac{|y_{1}y_{2}|}{y_{1}^{2} + y_{2}^{2}} \leq \frac{1}{2} \) (Binomial Theorem).

For all \( y_{1} \geq 1 \) by Lemma 13

\[
\sum_{y_{2}=0}^{\infty} \left( \frac{1}{y_{1}^{2} + y_{2}^{2}} \right) = \sum_{y_{2}=0}^{y_{1}} \left( \frac{1}{y_{1}^{2} + y_{2}^{2}} \right) + \sum_{y_{2} > y_{1}} \left( \frac{1}{y_{1}^{2} + y_{2}^{2}} \right) \\
\leq \sum_{y_{2}=0}^{y_{1}} \frac{1}{y_{1}^{2}} + \sum_{y_{2} > y_{1}} \frac{1}{y_{2}^{2}} \leq \frac{y_{1} + 1}{y_{1}^{3}} + \frac{1}{3 y_{1}^{3}} \\
= \frac{4y_{1} + 3}{3 y_{1}^{3}} \leq \frac{4y_{1} + 3y_{1}}{3 y_{1}^{3}} = \frac{7}{3 y_{1}^{3}}. \tag{12}
\]
Symmetry in the two dimensions and Equations (10), (11), and (12) yield

\[
\sum_{y \in \mathbb{Z}^2} \max_t \left| \frac{y_1 H(y, t)}{t} \right| < \sum_{y_1=1}^{\infty} \sum_{y_2=0}^{\infty} \frac{1024 y_1}{9 (y_1^2 + y_2^2)^2} < \sum_{y_1=1}^{\infty} \frac{266}{y_1^2}\] (13)

\[
\sum_{y \in \mathbb{Z}^2} \max_t \left| \frac{y_1 y_2 H(y, t)}{t^2} \right| < \sum_{y_1=1}^{\infty} \sum_{y_2=0}^{\infty} \frac{1024 (y_1^2 + y_2^2)^2}{(y_1^2 + y_2^2)^2} < \sum_{y_1=1}^{\infty} \frac{2390}{y_1^3}\] (14)

\(\sum y^{-2}\) and \(\sum y^{-3}\) from Equations (13) and (14), respectively, are well-known convergent series and can be calculated via the Riemann zeta function. As \(\sum_{y>0} y^{-2} = \zeta(2) = \pi^2/6\) and \(\sum_{y>0} y^{-3} = \zeta(3) \approx 1.202\) (Apéry’s constant), one gets:

\[
\sum_{y \in \mathbb{Z}^2} \max_t \left| \frac{y_1 H(y, t)}{t} \right| < \sum_{y_1=1}^{\infty} \frac{266}{y_1^2} = 266 \zeta(2) < 438,
\]

\[
\sum_{y \in \mathbb{Z}^2} \max_t \left| \frac{y_1 y_2 H(y, t)}{t^2} \right| < \sum_{y_1=1}^{\infty} \frac{2390}{y_1^3} = 2390 \zeta(3) < 2873.
\]

These bounds are much too coarse as an upper bound for the single vertex discrepancy. Therefore, we need to carefully sum up sufficiently many summands instead of bounding the equations of Lemma 7 directly, as we did above.

### 3.3 Computation

Aiming for an upper bound of the single vertex discrepancy we have, based on Lemma 1 derived Lemma 7 in Section 3.1. In Section 3.2 we have shown convergence of both series involved. However, for tight calculations, the estimates of Equations (13) and (14) are still too large, as discussed above. Therefore, we will sum up the small entries elementwise and bound the tails with the inequalities derived in the previous section. With \(t_{\text{max}}(y)\) as defined in Equation (6):

\[
\sum_{y \in \mathbb{Z}^2} \max_t \left| \frac{y_1 H(y, t)}{t} \right| = \sum_{\|y\|_{\infty} \leq 8000, y_1 \equiv y_2 (\text{mod } 2)} \left| \frac{y_1 H(y, t_{\text{max}}(y))}{t_{\text{max}}(y)} \right| + E_1 \quad (15)
\]

with

\[
E_1 := \sum_{\|y\|_{\infty} > 8000, y_1 \equiv y_2 (\text{mod } 2)} \left| \frac{y_1 H(y, t_{\text{max}}(y))}{t_{\text{max}}(y)} \right|.
\]
For the second sum, we obtain analogously (with $t'_{\max}(y)$ as defined in Equation (15)):  

$$
\sum_{y \in \mathbb{Z}^2} \max_{t} \left| \frac{y_1 y_2 H(y, t)}{t^2} \right| < \sum_{\|y\|_{\infty} \leq 3000, \ y_1 \equiv y_2 \ (\text{mod} \ 2)} \left| \frac{y_1 y_2 H(y, t'_{\max}(y))}{(t'_{\max}(y))^2} \right| + E_2. \quad (17)
$$

By only summing up over $y$ with $\|y\|_{\infty} \leq 3000$, we missed

$$
E_2 := \sum_{\|y\|_{\infty} > 3000, \ y_1 \equiv y_2 \ (\text{mod} \ 2)} \frac{y_1 y_2 H(y, t'_{\max}(y))}{(t'_{\max}(y))^2},
$$

which will be bounded in Equation (25) below.

For Equation (17), one calculates

$$
\sum_{\|y\|_{\infty} \leq 3000, \ y_1 \equiv y_2 \ (\text{mod} \ 2)} \left| \frac{y_1 y_2 H(y, t'_{\max}(y))}{(t'_{\max}(y))^2} \right| = 4 \sum_{y_1 = 1}^{3000} \sum_{y_2 \equiv y_1 \ (\text{mod} \ 2)} \frac{y_1 y_2 H(y, t'_{\max}(y))}{(t'_{\max}(y))^2}
$$

$$
= 4 \sum_{y_1 = 1}^{3000} \sum_{y_2 = 1}^{3000} \frac{y_1 y_2 4^{-t_{\max}(y)}}{(t'_{\max}(y))^2} \left( \frac{t_{\max}(y)}{(t_{\max}(y) + y_1)/2} \right) \left( \frac{t_{\max}(y)}{(t_{\max}(y) + y_2)/2} \right)
$$

$$
< 1.6439. \quad (18)
$$
To get exact upper bounds for $\sum \max_t |y_1 H(y, t)|$ and $\sum \max_t |y_2 H(y, t)|$, we need to upper bound the tail errors $E_1$ and $E_2$.

For $E_1$, we need to be very careful. In Equation (16) we only summed up over $|y_1|, |y_2| \leq 8000$. Hence, there are three kinds of summands in $E_1$: Such with $|y_1| \leq 8000$ and $|y_2| > 8000$ (Eq. 21), $|y_1| > 8000$ and $|y_2| \leq 8000$ (Eq. 23), and $|y_1|, |y_2| > 8000$ (Eq. 24). These three sums correspond to the yellow-shaded area shown in Figure 7 and to the three sums of Equation (19) below. Note that Equation (19) is an inequality, because on the right side we sum up over $y_1$ with $|y_1| > 8000$ and $y_2 = 0$ twice. This simplifies the calculation and overestimates $E_1$ by just $< 10^{-6}$.

We want to utilize Equation (10) from the previous section to estimate the three sums of Equation (19) above. This results in infinite sums over $\frac{y_1}{(y_1^2 + y_2^2)\pi}$, which can expressed as finite sums of the digamma function $\Psi_0$ and the first polygamma function $\Psi_1$. Fortunately, these infinite sums can be bounded tightly much easier via integral calculus. Hence, for $c \in \{0, 1\}$ and $\alpha, \gamma \in \mathbb{Z}$:

$$\sum_{\substack{y > \alpha, y \equiv c \mod 2}} \frac{1}{(y^2 + \gamma^2)^2} = \sum_{y > (\alpha - c)/2} \frac{1}{((2y + c)^2 + \gamma^2)^2} \leq \int_{y = (\alpha - c)/2}^{\infty} \frac{1}{((2y + c)^2 + \gamma^2)^2} = \left(\pi - 2 \arctan\left(\frac{\gamma}{\alpha}\right)\right) \frac{\alpha^2 + \gamma^2}{8(\alpha^2 + \gamma^2)^{3/2}} - 2\alpha\gamma.$$ (20)
Equipped with this, Equation (10) yields for the first sum of $E_1$:

$$
4 \sum_{y_1=1}^{8000} \sum_{y_2=8001, \ y_2 \equiv y_1 \pmod{2}}^{\infty} \frac{y_1 H(y, t_{\max}(y))}{t_{\max}(y)} < \frac{1024}{9} \sum_{y_1=1}^{8000} \sum_{y_2>8000, \ y_2 \equiv y_1 \pmod{2}} \frac{1}{(y_1^2 + y_2^2)^2}
$$

$$
\leq \frac{128}{9} \sum_{y_1=1}^{8000} \frac{(8000^2 + y_1^2)(\pi - 2 \arctan(8000/y_1)) - 16000 y_1}{(y_1^2 + 8000^2) y_1^2}.
$$

$$
< 0.0008.
$$

(21)

For the second sum of $E_1$, we need

$$
\sum_{y>y_\alpha, \ y \equiv c \pmod{2}} \frac{y}{(y^2 + \gamma^2)^2} = \sum_{y>(\alpha-c)/2} \frac{2y + c}{((2y + c)^2 + \gamma^2)^2}
$$

$$
\leq \int_{y=(\alpha-c)/2}^{\infty} \frac{2y + c}{((2y + c)^2 + \gamma^2)^2}
$$

$$
= \frac{1}{4(\alpha^2 + \gamma^2)}.
$$

(22)

With this, the second sum of $E_1$ can be bounded as follows:

$$
4 \sum_{y_1=8001}^{\infty} \sum_{y_2=0, \ y_2 \equiv y_1 \pmod{2}}^{8000} \frac{y_1 H(y, t_{\max}(y))}{t_{\max}(y)}
$$

$$
= 4 \sum_{y_2=0}^{8000} \sum_{y_1=8001, \ y_1 \equiv y_2 \pmod{2}}^{\infty} \frac{y_1 H(y, t_{\max}(y))}{t_{\max}(y)}
$$

$$
< \frac{1024}{9} \sum_{y_2=0}^{8000} \sum_{y_1>8000, \ y_1 \equiv y_2 \pmod{2}} \frac{y_1}{(y_1^2 + y_2^2)^2}
$$

$$
\leq \frac{256}{9} \sum_{y_2=0}^{8000} \frac{1}{y_2^2 + 8000^2}
$$

$$
< 0.0028.
$$

(23)
For the third sum of Equation (19), we need Equation (20) and
\[
\sum_{y > \beta} \frac{\pi - 2 \arctan\left(\frac{\alpha}{y}\right)}{y^2} \leq \int_{y = \beta}^{\infty} \frac{\pi - 2 \arctan\left(\frac{\alpha}{y}\right)}{y^2} = \frac{\ln(\alpha^2 + \beta^2) - 2 \ln(\beta)}{\alpha} + \frac{\pi - 2 \arctan\left(\frac{\alpha}{\beta}\right)}{\beta}
\]
\[
\sum_{y \geq \beta} \frac{1}{(\alpha^2 + y^2)y} \geq \int_{y = \beta}^{\infty} \frac{1}{(\alpha^2 + y^2)y} = \frac{\ln(\alpha^2 + \beta^2) - 2 \ln(\beta)}{2\alpha^2}.
\]
With this and Equations (10) and (20), we get for the third sum of \( E_1 \):
\[
4 \sum_{y_1 = 8001}^{\infty} \sum_{\substack{y_2 = 8001, \\ y_2 \equiv y_1 (\text{mod } 2)}} \frac{y_1 H(y, t_{max}(y))}{t_{max}(y)} \leq 1024 \sum_{y_1 > 8000} y_1 \sum_{\substack{y_2 > 8000, \\ y_2 \equiv y_1 (\text{mod } 2)}} \frac{1}{(y_1^2 + y_2^2)^2} \leq 128 \sum_{y_1 > 8000} \left( \frac{\pi - 2 \arctan\left(\frac{8000}{y_1}\right)}{(8000^2 + y_1^2)} - 16000y_1 \right) y_1 \leq 128 \left( \frac{\ln(2 \cdot 8000^2) - 2 \ln(8000)}{8000} + \frac{\pi/2}{8000} - \frac{\ln(2 \cdot 8001^2) - 2 \ln(8001)}{8001} \right) < 0.0028.
\]
The second error \( E_2 \) (cf. Equation (17)) is much easier to bound. With Equations (11) and (12), we get
\[
E_2 < 4 \sum_{y_1 = 1}^{3000} \sum_{y_2 > 3000} \frac{y_1 y_2 H(y, t'_{max}(y))}{(t'_{max}(y))^2} + 4 \sum_{y_1 > 3000} \sum_{y_2 > 0} \frac{y_1 y_2 H(y, t'_{max}(y))}{(t'_{max}(y))^2} < 4 \sum_{y_1 = 1}^{3000} \sum_{y_2 > 3000} \frac{1024}{(y_1^2 + y_2^2)^2} + 4 \sum_{y_1 > 3000} \sum_{y_2 \geq 0} \frac{1024}{(y_1^2 + y_2^2)^2} < 3000 \sum_{y_1 = 1}^{3000} \sum_{y_2 > 3000} \frac{4096}{y_2^4} + \sum_{y_1 > 3000} \frac{28672}{3y_1^4} \leq \frac{4096}{3 \cdot 3000^2} + \frac{14336}{3 \cdot 3000^2} < 0.0007.
\]
3. Upper Bound Computation

Now we can put all this together and plug the bounds obtained in Equations (16) and (19) in Equation (15) to get accurately

$$\sum_{y \in \mathbb{Z}^2} \max_t \left| \frac{y_1 H(y, t)}{t} \right| < 2.5248 + 0.0064$$

$$= 2.5312 \quad (26)$$

and analogously with the results of Equations (16) and (19) in Equation (15):

$$\sum_{y \in \mathbb{Z}^2} \max_t \left| \frac{y_1 y_2 H(y, t)}{t^2} \right| < 1.6439 + 0.0007$$

$$= 1.6446. \quad (27)$$

Combined with Lemma 7, this finally proves Theorem 6.
The configuration following the simple example shown in Figure 8 to the right already has a single vertex discrepancy of three at the central vertex. The linear machine walk moves one chip there, while the Propp machine moves all four chips to the central vertex. To get a better lower bound, one needs significantly more chips as well as stronger methods. The heart of the proof of the following main result of this chapter will be the Arrow-forcing Theorem 9 below.

**Theorem 8.** For clockwise $(\uparrow, \downarrow, \rightarrow, \leftarrow)$ and counterclockwise $(\downarrow, \uparrow, \rightarrow, \leftarrow)$ rotor sequences, there is an initial even configuration such that there is a time $t$ and a location $x$ with

$$|f(x, t) - E(x, t)| > 7.7.$$  

For all other rotor sequences, i.e., for $(\uparrow, \downarrow, \rightarrow, \leftarrow), (\downarrow, \uparrow, \leftarrow, \rightarrow), (\uparrow, \downarrow, \leftarrow, \rightarrow), (\downarrow, \uparrow, \rightarrow, \leftarrow), (\rightarrow, \leftarrow, \uparrow, \downarrow), (\leftarrow, \rightarrow, \downarrow, \uparrow)$, and $(\rightarrow, \leftarrow, \downarrow, \uparrow)$, there is an initial even configuration such that there is a time $t$ and a location $x$ with

$$|f(x, t) - E(x, t)| > 7.2.$$  

### 4.1 Arrow-forcing Theorem

To construct the desired configuration for the proof of Theorem 8 we need

**Theorem 9** (Arrow-forcing Theorem). Let $\rho(x, t) \in \text{DIR}$ be arbitrarily defined for $t \geq 0$ integer and $x \sim t$. Then there exists an even initial configuration that results in a game with $\text{ARR}(x, t) = \rho(x, t)$ for all such $x$ and $t$.  

---
4. Lower Bound

**Arrow-forcing Theorem**

![Diagram showing arrow-forcing]  

**Figure 9:** Basic step of the construction for the arrow sequence ($\nearrow, \searrow, \nearrow, \searrow$).

**Proof.** The proof does not depend on any specific rotor sequence since we only access it indirectly via the predicate NEXT. It also extends to the “free” version of the Propp machine described in Chapter 5.

The sought-after configuration can be found iteratively. We start by defining

\[ f(x, 0) := 0 \quad \text{for all } x \text{ (no chips anywhere)} \]

\[ \text{ARR}(x, 0) := \begin{cases} 
\rho(x, 0) & \text{for even } x \\
\rho(x, 1) & \text{for odd } x.
\end{cases} \]

This gives \( \text{ARR}(x, t) = \rho(x, t) \) on all \( x \sim t \) for \( t \leq 1 \). Now assume the functions \( f \) and \( \text{ARR} \) describe a game following an even initial configuration and there is a time \( T \geq 0 \) with \( \text{ARR}(x, t) = \rho(x, t) \) for all \( 0 \leq t \leq T + 1 \) and \( x \sim t \). The aim is to achieve \( \text{ARR}(x, t) = \rho(x, t) \) also for \( t = T + 2 \). Consider an \( x \) with \( \text{NEXT}(\text{ARR}(x, T + 2)) = \rho(x, T + 2) \). Such an \( x \) shall be the central position of the grids in Figure 9. There you see the neighborhoods of \( x \) at time steps \( T, T + 1, \) and \( T + 2 \). The black and gray dots mark positions \( y \) with \( y \sim t \) and \( y \sim (t + 1) \), respectively. Observe that, independent of the number of chips and the other arrows, the added chip on position \( x \) at time \( T \) changes nothing else on the black dots but the arrow of \( x \) at time \( T + 2 \) to \( \rho(x, T + 2) \). For

![Spreading out of a pile of 4^3 chips]  

**Figure 10:** Spreading out of a pile of \( 4^3 \) chips.
We need to ensure that the parities of the piles are the same.

The question remains, how we get the additional chips on a position $x$ at time $T$ without changing the arrows in the previous time steps. This can be done by adding multiples of $4^T$ chips (see Figure [11]) on the right positions at time 0 since a pile of $4^T$ chips will split evenly $T$ times so that the arrows at time $t \leq T$ remain the same.

Hence, we modify our initial configuration by setting

$$f'(x, 0) := \begin{cases} f(x, 0) + \epsilon_x 4^T & \text{for even } x \\ 0 & \text{for odd } x. \end{cases}$$

$$\text{ARR}'(x, 0) := \text{ARR}(x, 0) \quad \text{for all } x.$$

Here, the $\epsilon_x \in \{0, 1, 2, 3\}$ are to be determined. Our goal is to choose the values of $\epsilon_x$ so that $\text{ARR}'(x, t) = \rho(x, t)$ for all $t \leq T + 2$ and $x \sim t$. Due to the $T$ even splits of the piles of $4^T$ chips this holds automatically for $t \leq T$. Since we started with an even initial configuration there are no chips on $x \sim T + 1$ at time $T$. Hence, for $x \sim T + 1$ at time $T + 1$ we have $\text{ARR}'(x, T + 1) = \text{ARR}'(x, T) = \text{ARR}(x, T) = \text{ARR}(x, T + 1) = \rho(x, T + 1)$. To make sure the equality also holds for time $T + 2$ we need to ensure that the parities of the piles $f'(x, T)$ are right. Observe that $\text{ARR}'(x, T + 2) = \text{NEXT}^k(\text{ARR}'(x, T))$ for $f'(x, T) \equiv k \pmod{4}$. Hence, for $x \sim T$ we want $f'(x, T) \equiv k \pmod{4}$ exactly for $\rho(x, T + 2) = \text{NEXT}^k(\rho(x, T))$. For that, we watch closely how the “extra” groups of $4^T$ chips are spreading:

$$f'(x, 1) = f(x, 1) + 4^{T-1} \sum_{d \in \text{DIR}} \epsilon_{x+d}$$

(for odd $x$ and $T \geq 1$)

$$= f(x, 1) + 4^{T-1} \left( \epsilon_{x+(\frac{1}{1})} + \epsilon_{x+(\frac{1}{1})} + \epsilon_{x+(\frac{-1}{1})} + \epsilon_{x+(\frac{-1}{1})} \right)$$

$$f'(x, 2) = f(x, 2) + 4^{T-2} \sum_{d_1 \in \text{DIR}} \sum_{d_2 \in \text{DIR}} \epsilon_{x+d_1+d_2}$$

(for even $x$ and $T \geq 2$)

$$= f(x, 2) + 4^{T-2} \left( \epsilon_{x+(\frac{2}{1})} + \epsilon_{x+(\frac{2}{1})} + \epsilon_{x+(\frac{-2}{1})} + \epsilon_{x+(\frac{-2}{1})} \right)$$

$$+ 4^{T-1.5} \left( \epsilon_{x+(\frac{3}{1})} + \epsilon_{x+(\frac{3}{1})} + \epsilon_{x+(\frac{3}{1})} + \epsilon_{x+(\frac{3}{1})} \right)$$

$$+ 4^{T-1} \epsilon_x$$

$$\ldots$$

$$f'(x, T) = f(x, T) + \sum_{d_1 \in \text{DIR}} \sum_{d_2 \in \text{DIR}} \ldots \sum_{d_T \in \text{DIR}} \epsilon_{x+\sum_{i=1}^T d_i}$$

(for $x \sim T$)

$$= f(x, T) + \sum_{y=0}^{T} \sum_{\|y-x\|_{\infty} \leq T} \epsilon_y \left( \frac{T}{T+\epsilon_1-y_1} \right) \left( \frac{T}{T+\epsilon_2-y_2} \right).$$

33
4. Lower Bound

Arrow-forcing Theorem

The last sum describes the three-dimensional analogon of Pascal’s Triangle. For \( T = 0 \) the sum is \( \varepsilon_x \), hence, for all even \( x \) with \( \rho(x, 2) = \text{next}^k(\rho(x, 0)) \) we add \( k \) chips on \( x \), i.e., \( \varepsilon_x := k \), and get the aspired \( \text{ARR}'(x, t) = \rho(x, t) \) on all \( x \sim t \) for \( t \leq 2 \).

For \( T > 0 \) notice, that

\[
\sum_{y \sim 0 \atop \|y - x\|_\infty \leq T} \varepsilon_y \left( \frac{T}{2} + \frac{x_1 - y_1}{2} \right) \left( \frac{T}{2} + \frac{x_2 - y_2}{2} \right) = \varepsilon_{x + (0)_{T}} + \varepsilon_{x + (-T)} + \varepsilon_{x + (T)} + h,
\]

where \( h \) depends only on \( \varepsilon_y \) with \( x_1 - T \leq y_1 \leq x_1 + T \land x_2 - T \leq y_2 \leq x_2 + T \) or \( x_1 - T < y_1 < x_1 + T \land x_2 - T \leq y_2 \leq x_2 + T \). We now have to determine the \( \varepsilon_y \) sequentially for growing rectangles of side length \( \ell \), namely we define

\[
R_\ell := \left\{ (x_1, x_2) \mid |x_1| \leq \ell \land |x_2| \leq \ell \right\} \quad \text{and} \quad \partial R_\ell := \left\{ (x_1, x_2) \mid |x_1| = \ell \lor |x_2| = \ell \right\}.
\]

We start the inductive process with \( \varepsilon_y := 0 \) for all \( y \in R_T \setminus \{(0)_{T}\} \). For odd \( T \) we also set \( \varepsilon_{(0)_{T}} := 0 \) and for even \( T \) we define \( \varepsilon_{(0)_{T}} := k \) with \( \rho((0)_{0}, T + 2) = \text{next}^k(\rho((0)_{0}, T)) \). In both cases this yields \( \text{ARR}'(x, T + 2) = \rho(x, T + 2) \) on all \( x \sim T \) with \( x \in R_0 \).

Assume we have achieved \( \text{ARR}'(x, T + 2) = \rho(x, T + 2) \) on all \( x \sim T \) with \( x \in R_\ell \), and \( \varepsilon_y \) has been defined for all \( y \in R_{\ell + T} \). The next step is to define the \( \varepsilon_y \) for all even \( y \in \partial R_{\ell + T+1} \) to obtain \( \text{ARR}'(x, T + 2) = \rho(x, T + 2) \) also on all \( x \sim T \) with \( x \in \partial R_{\ell+1} \). Since we have \( 2(\ell + 1) \) equations, but \( 2(\ell + T + 1) \) many \( \varepsilon_y \), the \( \varepsilon_y \)'s are under-determined and can be assigned sequentially by walking around the rectangle \( \partial R_{\ell+T+1} \).

The described procedure changes an even initial configuration that matches the prescription in \( \rho \) for times \( t \leq T + 1 \) into another one that matches for times \( t \leq T + 2 \). Please note, that thereby we do not change the initial configuration of arrows \( \text{ARR}(x, 0) \) at all, and we change the initial number of chips \( f(x, 0) \) at position \( x \) only if \( \|x\|_\infty \geq T \). Hence, at all positions \( x \) the initial number of chips will be constant after the first \( \|x\|_\infty \) iterations. This shows that the process converges to an even initial configuration, which leads to the given game.
4.2 Computation

To get a large discrepancy at the origin, we choose via Theorem 9 above an initial configuration that sends as many chips as possible at the worst time $t_{\text{max}}(y)$ in the direction of the origin. This maximizes the sums of Lemma 1.

For clockwise rotor sequences, we send the chips as shown in Figure 11, i.e., one on both diagonals and two in the sectors between them. This is possible for $(\nearrow, \searrow, \nearrow, \searrow)$ and $(\searrow, \nearrow, \searrow, \nearrow)$, because $\searrow ((|y_1| < y_2 > 0)$, $\nearrow ((|y_1| < -y_2 > 0)$, $\nearrow (0 < y_1 > |y_2|)$, and $\searrow (0 < -y_1 > |y_2|)$ are consecutive rotor positions, which allows sending one chip in both directions at time $t_{\text{max}}(y)$. Let $x \neq 0$ be an arbitrary even position and $t_0 := t_{\text{max}}(x)$ as defined in Equation (6). For $(\nearrow, \searrow, \nearrow, \searrow)$ we apply the Arrow-forcing Theorem 9 to find an even starting position such that for all positions $y$ we have

$$\text{ARR}(y, t) := \begin{cases} 
\nearrow & \text{for all } 0 > y_1 \leq y_2 < -y_1 \text{ and } t \leq t_0 - t_{\text{max}}(y) \\
\searrow & \text{for all } 0 > -y_1 \leq y_2 < y_1 \text{ and } t > t_0 - t_{\text{max}}(y) \\
\nearrow & \text{for all } 0 > -y_2 \leq y_1 \leq y_2 \text{ and } t \leq t_0 - t_{\text{max}}(y) \\
\searrow & \text{for all } 0 > y_2 \leq y_1 < -y_2 \text{ and } t > t_0 - t_{\text{max}}(y) \\
\nearrow & \text{for all } 0 > y_1 \leq y_2 \leq y_1 \text{ and } t \leq t_0 - t_{\text{max}}(y) \\
\searrow & \text{for all } 0 > y_2 < y_1 \leq -y_2 \text{ and } t \leq t_0 - t_{\text{max}}(y) \\
\nearrow & \text{for all } 0 > -y_2 < y_1 \leq y_2 \text{ and } t > t_0 - t_{\text{max}}(y) \\
\searrow & \text{for all } 0 \leq y_1 = y_2.
\end{cases}$$

Note that in this case at a position $y$ with $\|y\|_2 \leq \|x\|_2$ we have a number of chips that is not a multiple of four exactly once at time $t_0 - t_{\text{max}}(y)$. For $y_1 = y_2$ there is exactly one chip at time $s_0(y) = t_0 - t_{\text{max}}(y)$, otherwise there are two chips at time $s_0(y) = s_1(y) = t_0 - t_{\text{max}}(y)$. The same holds true for the counterclockwise rotor sequence $(\searrow, \nearrow, \searrow, \nearrow)$ with a similar $\text{ARR}'(y, t)$.
Hence, with Lemma 1 we get for both clockwise and counterclockwise rotor sequences,

\[
f(0, t) - E(0, t) = \sum_{y \in \mathbb{Z}^2} \left( \sum_{i \geq 0} -A_1^{(i)}(y)y_1 H(y, t - s_i(y)) + \sum_{i \geq 0} -A_2^{(i)}(y)y_2 H(y, t - s_i(y)) \right)
+ \sum_{i \geq 0} \frac{A_1^{(i)}(y)A_2^{(i)}(y)y_1y_2}{(t - s_i(y))^2} H(y, t - s_i(y))
= \sum_{y \in \mathbb{Z}^2} \left( \sum_{i \geq 0} -A_1^{(i)}(y)y_1 H(y, t_{\text{max}}(y)) + \sum_{i \geq 0} -A_2^{(i)}(y)y_2 H(y, t_{\text{max}}(y)) \right)
+ \sum_{i \geq 0} \frac{A_1^{(i)}(y)A_2^{(i)}(y)y_1y_2}{(t_{\text{max}}(y))^2} H(y, t_{\text{max}}(y))
\]
\[
> 4 \sum_{y_1 = 0}^{3000} \sum_{y_2 = 1}^{y_2 - 1} \frac{2y_1}{t_{\text{max}}(y)} H(y, t_{\text{max}}(y))
+ 4 \sum_{y_1 = 0}^{3000} \frac{2y_1}{t_{\text{max}}(y_{y_1})} H(y, t_{\text{max}}(y_{y_1})) + \frac{y_1^2}{(t_{\text{max}}(y_{y_1}))^2} H(y, t_{\text{max}}(y_{y_1}))
+ 4 \sum_{y_1 = 0}^{3000} \frac{2y_2}{t_{\text{max}}(y)} H(y, t_{\text{max}}(y))
\]
\[
> 7.769.
\]

Since the clockwise and counterclockwise router sequences from above contained by definition all pairs of orthogonal rotors consecutively, we could send two chips in the direction of the origin per sector. In terms of Section 1.3 \( \chi_x(\pi) \neq (+, -, +, -) \) and \( \chi_y(\pi) \neq (+, -, +, -) \) for both (counter)clockwise permutations \( \pi \). Let us consider the two other rotor sequences with \( \chi_y(\pi) \neq (+, -, +, -) \), i.e., \( /, \backslash, \れ, \が \) and \( /, \し, \れ, \が \). They contain \( /, \し \), but not \( /, \れ \) or \( /, \が \) consecutively. Figure 12 shows how we send the chips at time \( t_{\text{max}}(y) \). The yellow shaded area shows the difference to Figure 11. There

![Figure 12: ARR(y,t) for rotor sequences (/,/, \れ, \が) and (/,/, \れ, \が).](image-url)
we can just send a single chip to the origin. Both remaining rotor sequences with $\chi_x(\pi) \neq (+, -, +, -)$, i.e., ($\leftarrow \rightarrow \leftarrow \leftarrow$, $\leftarrow 
rightarrow \leftarrow$) and ($\leftarrow \rightarrow \leftarrow \rightarrow$, $\leftarrow 
rightarrow \leftarrow$), can be handled analogously. They contain $\leftarrow \rightarrow$ and $\rightarrow \rightarrow$, but not $\leftarrow \rightarrow$ and $\rightarrow \leftarrow$ consecutively. Hence, with Lemma 1, we get for all four rotor sequences ($\leftarrow \rightarrow \leftarrow \rightarrow$, $\leftarrow \rightarrow \rightarrow \rightarrow$, $\leftarrow \leftarrow \rightarrow \rightarrow$, $\leftarrow \leftarrow \leftarrow \leftarrow$),

$$f(0, t) - E(0, t) > 4 \sum_{y_1=1}^{3000} \sum_{y_2 \equiv y_1 (\text{mod} 2)} \left( \frac{y_1}{t_{\text{max}}(y)} H(y, t_{\text{max}}(y)) + \frac{y_2}{t_{\text{max}}(y)} H(y, t_{\text{max}}(y)) \right)$$

$$+ 8 \sum_{y_1=1}^{3000} \sum_{y_2=y_1+2, y_2 \equiv y_1 (\text{mod} 2)} \frac{y_2}{t_{\text{max}}(y)} H(y, t_{\text{max}}(y))$$

$$+ 6 \sum_{y_1=2, y_1 \equiv 0 (\text{mod} 2)} \frac{y_1}{t_{\text{max}}(y_1)} H(\left( \begin{array}{c} y_1 \\ 0 \end{array} \right), t_{\text{max}}(\left( \begin{array}{c} y_1 \\ 0 \end{array} \right)))$$

$$> 7.223.$$

This proves Theorem 8.
Free Propp Machine

Apart from changing the arrows according to a fixed permutation as described in Section 1.3, one can also just require the arrows to change in such a way that the number of chips sent in each direction differs by at most a constant $\Delta$.

**Definition 7.** A rotor sequence $A = (A^{(0)}, A^{(1)}, A^{(2)}, \ldots)$ is called "$\Delta$-free" if

$$\max_{t \geq 0} \left( \max_{d \in \text{dir}} \left| \{i \leq t \mid A^{(i)} = d\} \right| - \min_{d \in \text{dir}} \left| \{i \leq t \mid A^{(i)} = d\} \right| \right) \leq \Delta.$$ 

**Definition 8.** A free Propp machine is a Propp machine as described in Section 1.2, but the rotor sequences of all vertices are $\Delta$-free for fixed $\Delta$.

The permutations of Section 1.3 are a special case of the free Propp machine with $\Delta = 1$. Note, that in contrast to the original Propp machine, here we are not distributing multiples of four chips equally to all four neighbors.

### 5.1 Upper Bound

For the discrepancy on a single vertex, we prove the following theorem.

**Theorem 10.** Independent of the initial even configuration, the time $t$, and the location $x$, we have

$$|f(x, t) - E(x, t)| < 26.9 \cdot \Delta.$$ 

For the proof of Theorem 10 we will need the following lemma.
Lemma 11. For every $\Delta$-free rotor sequence $A$, there is a $4\Delta$-coloring $\chi : \mathbb{N}_0 \rightarrow [1, 4\Delta]$ such that all monochromatic subsequences are alternating on the first dimension. This is optimal.

Proof. First, we show that for arbitrary $\Delta$-free rotor sequences there is no coloring with less colors. Consider the following $\Delta$-free rotor sequence:

$$A' := (\overbrace{\ldots}^{\Delta \text{ times}} \overbrace{\ldots}^{\Delta \text{ times}} \overbrace{\ldots}^{\Delta \text{ times}} \overbrace{\ldots}^{\Delta \text{ times}} \overbrace{\ldots}^{\Delta \text{ times}} \overbrace{\ldots}^{\Delta \text{ times}}).$$

One requires at least $4\Delta$ colors to separate alternating sequences on the first dimension:

$$A'_1 = (\overbrace{+}^{2\Delta \text{ times}} \overbrace{-}^{4\Delta \text{ times}} \ldots).$$

Now consider the first dimension $A_1^{(i)}$ of the given rotor sequence. For all $t \geq 0$:

$$\left| \sum_{i=0}^{t} A_1^{(i)} \right| = \left| \{i \leq t \mid A^{(i)} = \top\} + \{i \leq t \mid A^{(i)} = \bot\} - \{i \leq t \mid A^{(i)} = \bot\} - \{i \leq t \mid A^{(i)} = \top\} \right|$$

$$\leq \left| 2 \max_{d \in \text{dir}} \{i \leq t \mid A^{(i)} = d\} \right| - 2 \min_{d \in \text{dir}} \{i \leq t \mid A^{(i)} = d\} \right|$$

$$= 2\Delta. \quad (28)$$

Given any sequence $A_1^{(i)}$ of $\{-1, +1\}$ satisfying Equation (28), we now seek a $4\Delta$-coloring of $A_1^{(i)}$ such that all monochromatic subsequences are alternating sequences on their own. Our $4\Delta$ colors are called $\{-4\Delta + 1, -4\Delta + 3, \ldots, 4\Delta - 3, 4\Delta - 1\}$ for the time being. We color each $A_1^{(i)}$ with

$$\chi(t) := 2 \sum_{i=0}^{t-1} A_1^{(i)} + A_1^{(t)}.$$

This is a valid color for all $t$ since Equation (28) gives $|\sum_{i=0}^{t-1} A_1^{(i)}| + |\sum_{i=0}^{t} A_1^{(i)}| \leq 4\Delta$, which yields $|2 \sum_{i=0}^{t-1} A_1^{(i)} + A_1^{(t)}| \leq 4\Delta$ given that $\sum_{i=0}^{t-1} A_1^{(i)}$ and $\sum_{i=0}^{t} A_1^{(i)}$ cannot be of opposite sign. It remains to prove that all $4\Delta$ monochromatic subsequences are alternating. Between two consecutive items $A^{(k)}$ and $A^{(l)}$ of equal color, there are as many $+1$ as $-1$ on the first dimension, i.e., $\sum_{i=k+1}^{l} A_1^{(i)} = 0$. $\chi(k) = \chi(l)$ implies $2 \sum_{i=k}^{l-1} A_1^{(i)} + A_1^{(l)} = A_1^{(k)}$, thus $A_1^{(k)} + A_1^{(l)} = 0$ by $\sum_{i=k+1}^{l} A_1^{(i)} = 0$.

Note that these colorings are different from the colorings in Section 1.3.
5. Free Propp Machine

0. Hence, $A^{(k)}_1$ and $A^{(r)}_1$ are indeed of opposite sign and therefore $\chi(i)$ a valid coloring. Renumbering the colors gives the final coloring $\chi'(i) := (\chi(i) + 4\Delta + 1)/2$.

Proof of Theorem 10: The only difference between the original Propp machine introduced in Chapter 1 and the free Propp machine considered here, is the more general concept of the Δ-free rotor sequences according to Definition 7. In the proof of the upper bound of the original Propp machine (cf. Theorem 6), this becomes important when the three sums of Lemma 1 are decomposed in strictly alternating sums in Section 3.1.

We first study decompositions of $\sum_{i \geq 0} \frac{-A^{(i)}_1(y)y_1}{t-s_i(y)} H(y, t-s_i(y))$. In Section 3.1 this was easy: For two out of six rotor sequences, $A^{(i)}_1(y)$ was already alternating. For all other rotor sequences, we decomposed the sum into an alternating sum of the even and an alternating sum of the odd elements. For the free Propp machine, Lemma 11 gives us a $4\Delta$-coloring $\chi$ of the Δ-free rotor sequence such that all monochromatic subsequences are alternating on the first dimension. Now each subsum gets one color assigned:

$$\sum_{i \geq 0} \frac{-A^{(i)}_1(y)y_1}{t-s_i(y)} H(y, t-s_i(y)) = \sum_{c \leq 4\Delta} \sum_{i \geq 0, \chi(i) = c} \frac{-A^{(i)}_1(y)y_1}{t-s_i(y)} H(y, t-s_i(y)).$$

Analogously to Lemma 11 one can prove that there are $4\Delta$-colorings such that $A^{(i)}_1(y)$ or $A^{(i)}_1(y)A^{(2)}_i(y)$ are alternating. Hence, similar to Lemma 7, we have independent of the initial even configuration, the time $t$, and the location $x$,

$$f(x, t) - E(x, t) \leq \sum_{y \in \mathbb{Z}^2} \left(8\Delta \max_t \left| \frac{y_1 H(y, t)}{t} \right| + 4\Delta \max_t \left| \frac{y_1y_2 H(y, t)}{t^2} \right| \right).$$

Plugged in the constants from Equations (26) and (27), we finally get Theorem 10.
5.2 Lower Bound

Analogous to the introductory example of Chapter 4, the configuration following the right Figure 13 has discrepancy $3\Delta$ at the central vertex if the rotor sequence is suitably chosen. Fortunately, this concept of gluing $\Delta$ chips together scales well. Therefore, consider a classic Propp machine with rotor sequence $(\nearrow, \searrow, \nearrow, \searrow)$ and an initial configuration which has discrepancy $> 7.7$ on a vertex $x$ at time $t$. Such a machine exists according to Theorem 8. Out of this, we construct an initial configuration for a free Propp machine with rotor sequence

$$\left( \underbrace{\Delta \text{ times}}_{\Delta \text{ times}} \underbrace{\Delta \text{ times}}_{\Delta \text{ times}} \underbrace{\Delta \text{ times}}_{\Delta \text{ times}} \right).$$

On each vertex with $k$ chips in the original initial configuration, we now place a pile of $\Delta k$ chips. We also adopt the arrows by choosing the first arrow of the same direction within the rotor sequence for each location of the initial configuration. This results in a free Propp machine, which runs exactly as the original Propp machine, but with piles of $\Delta$ chips moving instead of single chips. This leads to a discrepancy of $7.7\Delta$ on $x$ at time $\Delta t$ and thus to the following theorem.

**Theorem 12.** For each $\Delta$, there is an initial even configuration and a rotor sequence such that there is a time $t$ and a location $x$ with

$$|f(x, t) - E(x, t)| \geq 7.7 \cdot \Delta.$$

This proof does not use that the free Propp machine might even send $2\Delta$ chips in one direction. For example the $\Delta$-free rotor sequence

$$\left( \underbrace{\Delta \text{ times}}_{\Delta \text{ times}} \underbrace{\Delta \text{ times}}_{\Delta \text{ times}} \underbrace{\Delta \text{ times}}_{\Delta \text{ times}} \right).$$

sends $2\Delta$ chips in the direction $\nearrow$. We expect this fact to help gaining a lower bound of $15\Delta$ for the single vertex discrepancy of the free Propp machine.
Chapter 6

Conclusions and Further Work

We examined the Propp machine on the two-dimensional infinite grid $\mathbb{Z}^2$. Independent of the starting configuration, at each time and on each vertex, the number of chips on a vertex deviates from the expected number of chips in the random walk model by at most a constant $c$. If the rotor is changing clockwise or counterclockwise, this can be bounded by $7.7 < c < 11.8$. For all other rotor sequences (cf. Section 1.3), this can be lowered to $7.2 < c < 10.9$. It remains to prove that both intervals are actually disjoint. However, to the best of our knowledge, this is the first work to indicate an influence of the rotor sequence on the discrepancy. This is also supported experimentally, as discussed in Section 1.4 for the aggregating Propp machine.

In addition, we analyzed a “free” variant of the Propp machine, where we only required the rotors to change in such a way that the number of chips sent in different directions differs by at most a constant $\Delta$. For this model, we bounded the single vertex discrepancy $c'$ by $7.7\Delta < c' < 27\Delta$. Also here, it remains to bridge the gap between the lower and upper bound. We expect the lower bound to be significantly higher.

We regarded as well the discrepancy on a single vertex over a time interval. For $x \in \mathbb{Z}^2$ and finite $S \subseteq \mathbb{N}_0$, we set analogously to Definition 3, $f(x, S) := \sum_{t \in S} f(x, t)$ and $E(x, S) := \sum_{t \in S} E(x, t)$. With Lemma 1 one can prove that the discrepancy of a single vertex $x$ in a time interval $S$ of length $T$ is at most $|f(x, S) - E(x, S)| = O(\sqrt{T \log T})$. It would also be interesting to examine the discrepancy on a rectangle or circle.
In higher dimensions, there is no equivalent to Lemma 1. Also note that in higher dimensions, there is no coordinate transformation, which results in all dimensions varying independently (cf. Assumption 2). However, we can analyze $\mathbb{Z}^d$ with another neighborhood definition. That is, we allow $2^d$ rotor directions $\{x \mid x_i \in \{-1, +1\}\}$ instead of the $2d$ rotor directions $\{x \mid \|x\|_1 = 1\}$ we had before. Since we still require Assumption 1 the probability that a chip from the origin arrives at $x$ at time $t$ in a simple random walk on $\mathbb{Z}^d$ is

$$H(x, t) = 2^{-dt} \prod_{i=1}^{d} \left( \frac{t}{(t + x_i)/2} \right)$$

for $x \sim t$ (cf. Definition 1). Analogously to Lemmas 4 and 5, one can show $\text{arg max}_t \left( H(x, t)/t^k \right) \approx \sum_{i=1}^{d} x_i^2/(d + 2k)$ for all $1 \leq k \leq d$. The higher dimensional equivalent to Lemma 1 is

$$f(0, t) - E(0, t) = \sum_{y \in \mathbb{Z}^d} \left( \sum_{i=1}^{d} (-1)^{i} A^{(d)}_{ij} \frac{H(y, t - s_i(y))}{H(y, t)} \right)$$. This can be used to calculate discrepancy bounds for these graphs.

Furthermore, one could look at rotors with biased directions. Propp [35] suggested to study the rotor sequences $\rightarrow \uparrow \leftarrow \rightarrow \uparrow \leftarrow \rightarrow \ldots$ or $\uparrow \rightarrow \uparrow \rightarrow \ldots$ on $\mathbb{Z}^2$. One could also try to use fewer rotors, e.g., to allow just the directions $\uparrow$ and $\downarrow$ if $i + j$ is even and $\leftarrow$ and $\rightarrow$ if $i + j$ is odd. We expect these Propp machines to behave very similar to the corresponding expected random walks.

Besides the just discussed graphs, it would be interesting to examine where the difference lies between a graph with and without constant single vertex discrepancy. It is know that finite graphs and infinite grids obey this “Propp property”. On the other hand, 3-regular trees seem to have unbounded discrepancy [14]. Further studies for other graph classes would be valuable.
Bibliography


Appendix A

Mathematical Basics

Lemma 13. \( \sum_{x>y} \frac{1}{x^k} \leq \frac{1}{(k - 1)y^{k-1}} \) for all \( y > 0 \) and all constants \( k > 1 \).

Proof. \( \sum_{x>y} \frac{1}{x^k} \leq \int_y^\infty \frac{1}{x^k} dx = \left[ \frac{1}{(1-k)x^{k-1}} \right]_{x=y}^{\infty} = \frac{1}{(k - 1)y^{k-1}}. \)

Lemma 14. \( \frac{1}{2} n^{-1/2} \leq 2^{-n} \left( \frac{n}{n/2} \right) \leq n^{-1/2} \) for \( n \geq 1 \).

Proof. From a sharp version of the Stirling formula \( \sqrt{2\pi n} \left( \frac{n}{e} \right)^n e\frac{1}{12n+1} < n! < \sqrt{2\pi n} \left( \frac{n}{e} \right)^n e\frac{1}{12n} \) due to Robbins [37] one gets \( (\sqrt{2\pi}^{-1} e^{-\frac{1}{12n+1}})n^{-1/2} < 2^{-n} \left( \frac{n}{n/2} \right) < (\sqrt{2\pi}^{-1} e^{-\frac{1}{12n(6n+1)}})n^{-1/2} \). With \( \frac{1}{2} < \sqrt{2\pi}^{-1} e^{-\frac{1}{3n(12n+1)}} \) and \( \sqrt{2\pi}^{-1} e^{-\frac{1}{12n(6n+1)}} < 1 \) for all \( n \geq 1 \) the lemma is shown. \( \square \)
Selbstständigkeitserklärung

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Jena, den 19. Dezember 2005

(Tobias Friedrich)