# The Computational Complexity of The Abelian Sandpile Model

Juan Andres Montoya, Carolina Mejia

June 2011

ABSTRACT This is the first volume of a series of studies related to the complexity theoretical analysis of statisitical mechanics. In this first work we have considered some predicting tasks arising in statistical mechanics. Predicting the evolution of dynamical systems is one of the foundational tasks of natural sciences. Statistical mechanics is mainly concerned with finite dynamical systems, which are more easy (from the conceptual point of view) of analyzing. We have chosen to work with one specific system of statisitical mechanics: The Abelian Sandpile Model.

# Contents

# Preface

Predicting the evolution of dynamical systems is one of the main tasks of natural science. Can we always predict? Chaos theory claims that there are dynamical systems which are unpredictable. In this work we are interested in analyzing the predictability of finite dynamical systems, which are predictable. We are interested in a strong notion of predictability: a dynamical system $\mathcal{S}$ is *strongly-predictable* if and only if we can predict the evolution of $\mathcal{S}$ without parsimoniously simulating its dynamics. We focus our attention on The Abelian Sandpile Model. Also, we consider the following technical question: Can we predict the evolution of a finite sandpile in polylogarithmic parallel time? or, do prediction problems related to The Abelian Sandpile Model belong to $NC$?

Our work is close related to the work of Moore and Machta (see for instance [MMG], [Mo]), which is concerned with the algorithmic complexity of simulating finite dynamical systems. In this work we study some prediction problems related to The Abelian Sandpile Model, which can be considered as a class of finite cellular automata (Boltzmann Systems). There are many works dealing with the algorithmic and physical complexity of Sandpile Prediction Problems (see for instance [BTW], [D], [M], [Mo]).

In this work we prove that prediction problems belong to $PSPACE$, we prove that the general prediction problem for Boltzmann Systems is $PSPACE$ complete, and we prove lower bounds for the restriction of The Sandpile Prediction Problem to low-dimensional lattices. Furthermore, we believe that we are defining the basis of a research project concerned with the analysis of the short term dynamics of finite cellular automata.

## 0.1 Organization of the work

## 0.2 Acknowledgements

Authors would like to thanks all the people that contribute with the writing of this booklet. We apologize if we forget to mention someone of our co-workers, friends, students and supporters. We want to thanks the graduate students that have taken part in our seminar (Computational Complexity of Statistical Mechanics), special thanks go to Sergio Andres Montoya, Francisco Javier Gutierrez and Sterling Castañeda. Thanks go also to our co-workers Carlos Arturo Rodriguez and Rafael Isaacs, and to our sandpile-friend Anahi Gajardo. This work would not see the light of day without the financial support of VIE-UIS and Colciencias research project 111518925292.

*Bucaramanga, Enero de 2011.*

# 1
# Basics

In this chapter we introduce some of the notation that will be used along the work, and some of the mathematical concepts that we will use in some places of this work. Furthermore, we will introduce some basic facts, and some basic concepts of complexity theory. We will focus our attention on parallel complexity, which provide us with a conceptual machinery that can be used to analyze the algorithmic hardness of most sandpile prediction problems.

## 1.1 Lattices

Lattice graphs are discrete versions of the euclidean space, because of this they have played an important role in statistical mechanics: lattices are the underlying graphs of most of the graphical models of statistical mechanics. We will focus our attention on the restriction of the abelian sandpile model to low-dimensional lattices, specifically we will consider the restriction of the abelian sandpile model to linear lattices, square lattices and cubic lattices (one-dimensional, two-dimensional and three-dimensional cubic lattices).

Given $n \geq 1$, we use the symbol $\mathcal{G}_n^1$ to denote the *linear lattice* of order $n$, whose vertex set is equal to $[n]$, where $[n] = \{1, ..., n\}$ and the edge relation is the nearest neighbor relation. We use the symbol $\mathcal{L}_n^1$ to denote the *linear sandpile lattice* of order $n$, which is obtained from $\mathcal{G}_n^1$ by adding a node $s$ which is called the *sink.* Furthermore, given $v$, a node on the border of $\mathcal{G}_n^1$, there are one edge in $\mathcal{L}_n^1$ connecting $v$ and $s$. Note that given $v \in V\left(\mathcal{L}_n^1\right) - \{s\}$, we have that $\deg(v) = 2$. We use the symbol $\mathcal{L}_1$ to denote the class $\left\{\mathcal{L}_n^1 : n \geq 1\right\}$.

We use the symbol $\mathcal{G}_n^2$ to denote the *square lattice* of order $n$, whose vertex set is equal to $[n] \times [n]$. We use the symbol $\mathcal{L}_n^2$ to denote the *square sandpile lattice* of order $n$, which is obtained from $\mathcal{G}_n^2$ by adding a node $s$ which is called the sink. Furthermore, given $v$ a node on the border of $\mathcal{G}_n^2$, there are $4 - \deg_{\mathcal{G}_n^2}(v)$ edges in $\mathcal{L}_n^2$ connecting $v$ and $s$. Note that given $v \in V\left(\mathcal{L}_n^2\right) - \{s\}$, we have that $\deg(v) = 4$. We use the symbol $\mathcal{L}_2$ to denote the bounded class $\left\{\mathcal{L}_n^2 : n \geq 1\right\}$.

Finally, we use the symbol $\mathcal{G}_n^3$ to denote the *cubic lattice* of order $n$, whose vertex set is equal to $[n] \times [n] \times [n]$. We use the symbol $\mathcal{L}_n^3$ to denote the *cubic sandpile lattice* of order $n$, which is obtained from $\mathcal{G}_n^3$ by adding a node $s$ called the sink. Furthermore, given $v$ a node on the border of $\mathcal{G}_n^3$, there are $6 - \deg_{\mathcal{G}_n^3}(v)$ edges in $\mathcal{L}_n^3$ connecting $v$ and $s$. Note that given $v \in V\left(\mathcal{L}_n^3\right) - \{s\}$, we have that $\deg(v) = 6$. We use the symbol $\mathcal{L}_3$ to denote the bounded class $\left\{\mathcal{L}_n^3 : n \geq 1\right\}$

## 1.2 Complexity theory

In this section we introduce some basic facts (and concepts) of Complexity Theory. Complexity Theory analyzes algorithmic problems with respect to their intrinsic hardness. The final goal of Complexity Theory is to determine the exact amount of computational resources required to solve a given problem. A very good introduction to the field is the reference [P].

### *1.2.1 Parallel complexity*

We will analyze problems that can be solved in polynomial time. Let $L$ be a computational problem, knowing that $L$ is polynomial time solvable is not the last word, some additional questions can be stated: Can $L$ be solved using logarithmic space? Can $L$ be solved in polylogarithmic time using a polynomial number of processors? Actually, these are the questions that we consider when analyzing the predicting tasks associated to The Abelian Sandpile Model. Because of this, we want to use this preliminary chapter to introduce the basic ideas of Parallel Complexity, which is a complexity theory well suited for dealing with the kind of questions we are considering.

First at all we introduce the classes $NC^i$, where $i$ belongs to $\mathbb{N}$. These classes can be introduced via uniform families of circuits with some specific constraints related to size, depth, fan in and fan out. We will introduce those classes from a pragmatic point of view as the classes of problems that can be solved in polylogarithmic time using a polynomial number of processors.

**Definition 1** *Given $i \geq 1$ we have that $NC^i$ is the class of problems that can be solved in time $O\left(\log^i(n)\right)$ using a polynomial number of processors.*

A fundamental notion of parallel complexity is the notion of $NC$-reduction, intuitively $NC$ reductions are Turing reductions that can be computed in polylogarithmic time.

**Definition 2** *Given $L$ and $\Pi$ two problems, we say that $L$ is $NC^i$ reducible to $\Pi$ if and only if there exists a Turing reduction of $L$ in $\Pi$ which can be computed in time $O\left(\log^i(n)\right)$ using a polynomial number of processors.*

There are some other *polylogarithmic classes* which will play an important role in our work.

**Definition 3** *(further polylogarithmic classes)*

1. *$L$ is the class of problems that can be solved using a logarithmic space bounded deterministic Turing Machine.*
2. *$NL$ is the class of problems that can be solved using a logarithmic space bounded nondeterministic Turing Machine.*
3. *A logspace reduction is a Turing reduction which can be computed using logarithmic space.*
4. *$logDCFL$ is the class of problems which are logspace reducible to a deterministic context free language.*
5. *$TC^0$ is the class of problems which can be solved using a polynomial size uniform family of circuits of bounded depth defined on the logical basis $\{\wedge, \vee, \sim\}$.*
6. *We say that $L$ is $TC^0$ hard if and only if the majority function is constant depth reducible to $L$, that is: $L$ is $TC^0$ hard if and only if there exists a reduction of $Maj$, the problem consisting in computing the majority function, in $L$ which can be computed using a polynomial size uniform family of circuits of constant depth.*

**Remark 4** *The definition of $TC^0$ hardness is based on Hastad-Sipser theorem, which says that $Maj$ doesn't belong to $TC^0$. Note that given $L$ an algorithmic problem, if $L$ is $TC^0$ hard we have that $L \notin TC^0$, the problem $L$ requires unbounded depth.*

It is known that

$$TC^0 \subset NC^1 \subset L \subset NL \subset logDCFL \subset NC^2 \subset ... \subset NC^i \subset NC^{i+1} \subset ... \subset P$$

Also, we have a hierarchy of complexity classes, and our goal is to find the right place, within this hierarchy, of each one of the problems that we want to analyze.

A last concept is the important concept of $P$-completeness. Intuitively, a $P$-complete problem is a *ptime* solvable problem which is very hard to be solvable in polylogarithmic parallel time.

**Definition 5** *Given $L \in P$, we say that $L$ is $P$-complete if and only if given $\Psi \in P$ there exists a $NC$ reduction of $\Psi$ in $L$.*

**Teorema 6** *Let $NC = \bigcup_{i \geq 1} NC^i$, and let $L$ be a $P$-complete problem. We have that $NC = P$ if and only if $L \in NC$.*

The proof of last theorem is straightforward. Last theorem says that if $L$ is $P$-complete, then it is very unlikely that $L \in NC$, since it is very unlikely that $P = NC$.

Let $MCVP$ the problem defined by

**Problem 7** *($MCVP$; monotone circuit value problem)*

- *Input: $(\mathcal{C}, \mu)$, where $\mathcal{C}$ is a monotone, synchronous boolean circuit of fan in $2$ and fan out $2$.*
- *Problem: Decide if $\mathcal{C}(\mu) = 1$.*

**Teorema 8** *(Cook's Reduction)*
*The problem $MCVP$ is $P$-complete.*

A proof of last theorem can be found in [P]. It is very easy to prove that given $L, \Pi \in P$, if $L$ is $P$-complete and $L$ is $NC$ reducible to $\Pi$, then $\Pi$ is $P$-complete. Also, the problem $MCVP$ can be used (and actually it is used) as a pivot in $P$-completeness proofs.

## 1.3 Exercises

1. Prove theorem 6.
2. Prove that $MCVP$ is $P$-complete.
3. Let $L$ be a problem in $P$. Prove that if $MCVP$ is $NC$ reducible to $L$, then $L$ is $P$-complete.

# 2
# The Complexity of Predicting

Finite dynamical systems are predictable, because the evolution operator is computable. Given $\mathcal{S}$ a finite dynamical system, given $f$ an initial configuration and given $t$, we can compute the configuration reached by $\mathcal{S}$ at time $t$, without observing the evolution of $\mathcal{S}$. The important question is the following one: Can we predict the configuration reached by $\mathcal{S}$ at time $t$ in less than $t$ time units? or, can we predict the configuration reached by $\mathcal{S}$ at time $t$ in $O\left(\log^{i}(t)\right)$ time units for some $i \geq 1$?

## 2.1 Prediction problems and Boltzmann systems

Our aim is to analyze the computational complexity of some prediction tasks related to the dynamics of some classes of finite dynamical systems. To begin, we define the dynamical systems that we want to study. We call *Boltzmann Systems* to those systems

A Boltzmann system is a triple $(G, Q, T)$ such that

- $G$ is a finite digraph.
- $Q$ is a finite set.
- $T : Q^{V(G)} \rightarrow Q^{V(G)}$ is a function which satisfies the following (continuity) condition

  Given $v \in V(G)$ and given $f_1, f_2 \in Q^{V(G)}$

$$\text{iff } f_1 \restriction_{N(v)} = f_2 \restriction_{N(v)}, \text{then } \left(T\left(f_1\right)\right)(v) = \left(T\left(f_1\right)\right)(v)$$

where $N(v)$ denotes the neighborhood of $v$. The elements of $Q^{V(G)}$ are called configurations.

Let $\mathcal{C}$ be a class of Boltzmann systems. We use the symbol $pred[\mathcal{C}]$ to denote the prediction problem defined by.

**Problem 9** *(The Prediction Problem)*

- *Input:* $(\mathcal{M}, f, t)$, *where* $\mathcal{M} = (G, Q, T) \in \mathcal{C}$; $f \in Q^{V(G)}$ *is a configuration and* $t \geq 0$.
- *Problem: Compute the configuration reached by* $\mathcal{M}$ *after* $t$ *iterations of the operator* $T$.

Suppose that $(G, Q, T)$ is a Boltzmann system. Note that the set $Q^{V(G)}$ of possible configurations has a size which is upperbounded by $|Q|^{|V(G)|}$. It implies that after at most $|Q|^{|V(G)|}$ iterations the system either reachs a fixed point or enters into a periodic trajectory. We can classify the classes of Boltzmann systems into two types: fixed point classes and periodic classes. So, we can consider two types of prediction problems. If $\mathcal{C}$ is a class of fixed point systems, given $(G, Q, T) \in \mathcal{C}$, given $f$ a configuration and given $t \gneqq |Q|^{|V(G)|}$ we have $T^{(t)}(f) = T^{\left(|Q|^{|V(G)|}\right)}(f)$. We say that $T^{\left(|Q|^{|V(G)|}\right)}(f)$ is the fixed point of $f$. Because of this, if we are dealing with a class of fixed point systems we can focus our attention on the following problem: let $\mathcal{C}$ be a class of fixed point systems

**Problem 10** *($FSP[\mathcal{C}]$, Final State Prediction)*

- *Input:* $((G, Q, T), f)$, *where* $(G, Q, T) \in \mathcal{C}$ *and* $f$ *is a configuration.*
- *Problem: Computes the fixed point of* $f$.

If $\mathcal{C}$ is not a fixed point class but a periodic class, we can to restrict our attention to small values of $t$, that is: we restrict the problem $pred[\mathcal{C}]$ to instances $((G, Q, T), f, t)$ such that $t \leq |Q|^{|V(G)|}$.

**Problem 11** *($PSP[\mathcal{C}]$, Periodic Systems Prediction)*

- *Input:* $((G, Q, T), f, t)$, *where* $((G, Q, T), f, t) \in \mathcal{C}$ *and* $t \leq |Q|^{|V(G)|}$.
- *Problem: Computes the configuration* $T^{(t)}(f)$.

## 2.2 Prediction problems and PSPACE

In this section we study the complexity of $FSP[\mathcal{C}]$ for some classes $\mathcal{C}$ of fixed point systems. First at all we establish the $PSPACE$ completeness of the problem $FSP$, ($FSP$ denotes the problem $FSP[\mathcal{C}]$ with $\mathcal{C}$ equal to the class of all the fixed point Boltzmann systems).

**Teorema 12** *$FSP$ is $PSPACE$ complete.*

**Proof.** First at all we have to define the way by which we measure the size of $FSP's$ instances. Let $(G,Q,T)$ an instance of $FSP$. We define $|(G,Q,T)|$, the size of $(G,Q,T)$, as $|V(G)|\,|Q|$, where $V(G)$ is the set of nodes of $G$. Note that we can codify a configuration $f$ on $(G,Q,T)$ as a vector $v_f$ such that $v_f$ has $|V(G)|$ entries and each one of its entries corresponds to a number in the interval $\{1,...,|Q|\}$. So, we can codify the configuration $f$ using $O(|V(G)|\log(|Q|))$ space. It is clear that we can compute the fixed point of $f$ by simulating the evolution of $(G,Q,T)$, when we begin with the configuration $f$. To this end we can use a naive simulation algorithm which saves, at any stage, the actual configuration (i.e. at any stage of the computation we save one and only one configuration: the last configuration reached by the system $(G,Q,T)$). Also, we can solve $FSP$ using $O(|V(G)|\log(|Q|))$ space (using polynomial space).

Let $\mathcal{M}$ be a polynomial space bounded Turing machine and let $x = x_1...x_n$ be an input of $\mathcal{M}$. We suppose that $\mathcal{M}$ is a decision machine. We can think of the pair $(\mathcal{M},x)$ as a Boltzmann system $(G_{\mathcal{M},x}, Q_{\mathcal{M},x}, T_{\mathcal{M},x})$ in the following way:

Suppose that $p(X)$ is a polynomial such that for any input $x$ of size $n$, the machine $\mathcal{M}$ uses at most $p(n)$ work cells. We define $(G_{\mathcal{M},x}, Q_{\mathcal{M},x}, T_{\mathcal{M},x})$ as follows

- $G_{\mathcal{M},x}$ is a graph constituted by two connected components, the first one, which we call $I_{\mathcal{M},x}$, is the linear graph with universe $\{1,...,n\}$, the second one, which we call $W_{\mathcal{M},x}$, is the linear graph with universe $\{1,...,p(n)\}$.
- $Q_{\mathcal{M},x,} = \Sigma_{\mathcal{M}} \times \{0,1\} \times Q_{\mathcal{M}}$, where $\Sigma_{\mathcal{M}}$ is the alphabet of $\mathcal{M}$ and $Q_{\mathcal{M}}$ is the set of inner states of $\mathcal{M}$.

Now we define the notion of feasible configuration. Given $f \in Q_{\mathcal{M},x}^{V(G_{\mathcal{M},x})}$ it is a feasible configuration if and only if:

1. For any $i \in I_{\mathcal{M},x}$ we have that $f(i) = (x_i, \epsilon, q)$, with $\epsilon \in \{0,1\}$.
2. There is exactly one $i \in I_{\mathcal{M},x}$ such that $\pi_2(f(i)) = 1$.
3. There exists exactly one $j \in W_{\mathcal{M},x}$ such that $\pi_2(f(j)) = 1$.
4. There exists $q \in Q_{\mathcal{M}}$ such that for any $i \in I_{\mathcal{M},x} \cup W_{\mathcal{M},x}$, we have that $\pi_3(f(i)) = q$.

Note that the set of feasible configurations of $(G_{\mathcal{M},x}, Q_{\mathcal{M},x}, T_{\mathcal{M},x})$ corresponds to the set of configurations which can be reached by $\mathcal{M}$ during its computation on input $x$.

- The definition of $T_{\mathcal{M},x}$ depends on the definition of $\delta_{\mathcal{M}} : Q_{\mathcal{M}} \times \Sigma_{\mathcal{M}} \times \Sigma_{\mathcal{M}} \rightarrow Q_{\mathcal{M}} \times \Sigma_{\mathcal{M}} \times \{\rightarrow, \leftarrow, \diamondsuit\} \times \{\rightarrow, \leftarrow, \diamondsuit\}$, the transition function of $\mathcal{M}$. Let $f = (v_1, ..., v_n, w_1, ..., w_m)$ be a feasible configuration and let $i$ and $j$ be the positions for which the second component of the pairs $v_i$ and $w_j$ is equal to 1. $T_{\mathcal{M},x}(f) = (v_1^*, ..., v_n^*, w_1^*, ..., w_m^*)$ is defined as follows:

1. For any $k \in I_{\mathcal{M},x}$ we have that $\pi_1(v_k^*) = x_k$.
2. Given $k \in I_{\mathcal{M},x}$ we have that $\pi_2(v_k^*) = 1$ if and only if either $k = i - 1$ and
$$\pi_3(\delta_{\mathcal{M}}(\pi_3(v_1), \pi_1(v_i), \pi_1(w_j))) = \leftarrow$$
or $k = i$ and
$$\pi_3(\delta_{\mathcal{M}}(\pi_3(v_1), \pi_1(v_i), \pi_1(w_j))) = \diamondsuit$$
or $k = i + 1$ and
$$\pi_3(\delta_{\mathcal{M}}(\pi_3(v_1), \pi_1(v_i), \pi_1(w_j))) = \rightarrow .$$
3. Given $k \in W_{\mathcal{M},x}$ we have that $\pi_2(w_k^*) = 1$ if and only if either $k = j - 1$ and
$$\pi_4(\delta_{\mathcal{M}}(\pi_3(v_1), \pi_1(v_i), \pi_1(w_j))) = \leftarrow$$
or $k = i$ and
$$\pi_4(\delta_{\mathcal{M}}(\pi_3(v_1), \pi_1(v_i), \pi_1(w_j))) = \diamondsuit$$
or $k = i + 1$ and
$$\pi_4(\delta_{\mathcal{M}}(\pi_3(v_1), \pi_1(v_i), \pi_1(w_j))) = \rightarrow$$
4. $\pi_1(w_j^*) = \pi_2(\delta_{\mathcal{M}}(\pi_3(v_1), \pi_1(v_i), \pi_1(w_j)))$.
5. For any $k \in I_{\mathcal{M},x}$ we have that
$$\pi_3(v_k^*) = \pi_1(\delta_{\mathcal{M}}(\pi_3(v_1), \pi_1(v_i), \pi_1(w_j)))$$
6. For any $k \in W_{\mathcal{M},x}$ we have that
$$\pi_3(w_k^*) = \pi_1(\delta_{\mathcal{M}}(\pi_3(v_1), \pi_1(v_i), \pi_1(w_j)))$$

Consider the algorithmic problem

- *Input:* $(\mathcal{M}, x)$, *where* $\mathcal{M}$ *is a polynomial space bounded Turing Machine and* $x$ *is an input of* $\mathcal{M}$.
- *Problem: Decides if* $\mathcal{M}$ *accepts* $x$.

It is clear that last problem is $PSPACE$ hard. Now we will show that we can can reduce this problem to $FSP$. The reduction is defined in the following way.

Given $(\mathcal{M}, x)$, we compute $((G_{\mathcal{M},x}, Q_{\mathcal{M},x}, T_{\mathcal{M},x}), f_x)$, where $f_x$ is the feasible configuration determined by $x$, then we compute $FP(f_x)$ the fixed point of $f_x$. Suppose that $FP(f_x) = (v_1, ..., v_n, w_1, ..., w_m)$, we compute $\pi_3(v_1)$. If $\pi_3(v_1)$ is an accepting state of $\mathcal{M}$ we accept $(\mathcal{M}, x)$, otherwise we reject.

It is easy to verify that last reduction is sound and polynomial time bounded. Therefore we have that $FSP$ is $PSPACE$ hard ■

**Remark 13** *Note that for any periodic class* $\mathcal{C}$ *the problem* $PSP[\mathcal{C}]$ *belongs to* $PSPACE$. *It implies that PSP is* $PSPACE$ *complete, since* $FSP$ *is a subproblem of PSP. Note that any fixed point system is a periodic system and given* $((G, Q, T), f)$ *an instance of* $FSP$, *it corresponds to the instance* $\left(((G, Q, T), f, |Q|^{|V(G)|}\right)$ *of PSP.*

**Remark 14** *It is important to remark that last reduction, which is a logspace reduction, can be adapted to prove that given* $\mathcal{C} \subseteq PSPACE$ *a complexity class containing a complete problem, there exists a class* $\mathcal{D}$ *of Boltzmann systems such that* $FSP[\mathcal{D}]$ *is* $\mathcal{C}$ *complete.*

One of the main topics of our research is the following one: Given $\mathcal{C}$ a fixed point class, can we solve the problem $FSP[\mathcal{C}]$ in polylogarithmic parallel time? We know that, it is not always possible to solve $FSP[\mathcal{C}]$ in polylogarithmic parallel time.

**Teorema 15** $FSP$ *doesn´t belongs to NC.*

**Proof.** We know that $FSP$ is $PSPACE$ complete, let $f$ be a sublinear function (i.e. $n \notin O(f)$), The Space hierarchy Theorem implies that $FSP \notin SPACE(f)$, also we have that $FSP \notin NC$ ■

Some of the most interesting (and applicable) prediction problems belong to $P$, and some of them belongs to $NC$. Let $\mathcal{PD}$ be the class of pushdown automata, and let $FSP[\mathcal{PD}]$ be the algorithmic problem defined by

**Problem 16** *(*$FSP[\mathcal{PD}]$*; predicting Pushdown Automata)*

- *Input:* $(\mathcal{M}, x)$, *where* $\mathcal{M}$ *is a pushdown automaton and* $x$ *is an input of* $\mathcal{M}$.
- *Problem: compute the final state of the computation of* $\mathcal{M}$ *on input* $x$.

**Teorema 17** *The problem $FSP[\mathcal{PD}]$ can be solved in time $O\left(\log^2(n)\right)$ using a polynomial number of processors.*

**Proof.** First at all we note that in order to compute the final state of the system $(\mathcal{M}, x)$ it is sufficient to decide if $\mathcal{M}$ accepts $x$. It is the case, since we know that, at the end of the computation the head of the pushdown tape will be placed at the right end of the tape, the content of the tape will be equal to the content at the beginning of the computation and the pushdown stack will become empty. Also, in order to determine the final configuration of the system it is sufficient to determine its final state, that is: it is sufficient to determine if the automaton $\mathcal{M}$ accepts $x$. This problem (determining if $x \in L(\mathcal{M})$) is a subproblem of the parsing problem for $L(\mathcal{M})$, which can be solved in time $O\left(\log^2(n)\right)$ using a polynomial number of processors. ■

The theorems above show that there are predictable (polylogarithmic time predictable) an unpredictable classes of Boltzmann systems. We are interested in classes of Boltzmann systems coming from statistical mechanics. There are many interesting examples of Boltzmann systems coming from statistical mechanics, some of them are The Abelian Sandpile Model [BTW], The Eulerian Walkers Model and Langton's Ant [GGM], The Ising Automaton [GM] and The Flipping Ising Dynamics *[Mo]*. In the following we will analyze The complexity of prediction problems associated to The Abelian Sandpile Model, which is the toy model of Self-organized Criticality and which is our toy model for the development of a complexity theoretical analysis of prediction problems.

## 2.3 Exercises

1. Prove that the prediction problem for linear bounded automata is $PSPACE$ complete.
2. Define a class of Boltzmann systems such that the prediction problem associated to it is $NC$-computable.
3. Define a class of Boltzmann systems such that the prediction problem associated to it is $NP$-complete.

# 3 The Abelian Sandpile Model

In this section we introduce the basic definitions and some basic facts concerning The Abelian Sandpile Model.

**Definition 18** *A sandpile graph is a pair* $(G, s)$, *where* $G$ *is a connected graph and* $s \in V(G)$.

Given $(G, s)$ a sandpile graph, the node $s$ will be called the *sink.* Most of the time we will say that $G$ is a sandpile graph and that $s$ is the sink of $G$. From now on, we will use the symbol $G$ to denote the pair $(G, s)$. The symbol $V(G)^*$ will denote the set $V(G) - \{s\}$. A *configuration* on $G$ is a function $g : V(G)^* \rightarrow \mathbb{N}$. Given $g$ a configuration on $G$ and given $v \in V(G)^*$ we will say that $v$ is *g-stable* if and only if $g(v) \lneqq \deg(v)$, and we will say that $g$ is an *stable configuration* if and only if for all $v \in V(G)^*$, we have that $v$ is $g$-stable.

**Definition 19** *Given* $G$ *a sandpile graph, the sandpile automaton on* $G$ *is the graph automaton* $\mathcal{SP}(G)$ *defined by*

1. *The set of configurations of* $\mathcal{SP}(G)$ *is the set*

$$\{g : g \text{ is a configuration on } G\}$$

2. *Given* $g$ *a configuration of* $\mathcal{SP}(G)$ *and given* $v$ *a cell, the state of* $v$ *under* $g$ *is equal to* $g(v)$.

3. *Given* $g$ *a configuration, the set of possible transitions from* $g$ *is given by the following transition rule:*

*Given $v \in V(G)^*$, if $g(v) \geq \deg(v)$, then we have that $g \to g_v$ is a possible transition, where $g_v$ is the configuration on $G$ defined by*

$$g_v(w) := \begin{cases} g(v) - \deg(v), \text{ if } w = v \\ g(w) + 1, \text{ if } v \text{ is a neighbor ancestor of } w \\ g(w) \text{ if } v \text{ is not a neighbor of } w \end{cases}$$

Any transition of $\mathcal{SP}(G)$ is called a *firing* or a *toppling.* So, given $g$ a configuration, the transition $g \to g_v$ is a firing, and if such a transition occurs, we say that the node $v$ was fired (toppled) or we say that a firing (toppling) at $v$ has occurred.

Given $G$ a sandpile graph and given $g$ an initial configuration, we can choose an unstable node, fire it and obtain a new configuration. Note that we can choose any unstable node to produce a firing, in this sense sandpile automata are nondeterministic. A sequence of firings $g_1 \to g_2 \to ... \to g_n$ is called an *avalanche* of length $n$ with initial configuration $g$, and we say that it is an avalanche from $g$ to $g_n$. If $g_n$ is stable we say that $g_n$ is a *stabilization* or a *relaxation* of $g$. If we fix a configuration $g$ on $V$, we can consider the following three sets: $Aval(G, g)$, the set of avalanches whose initial configuration is $g$; $Aval_M(G, g)$ the set of maximal avalanches beginning in $g$ ($A$ is maximal if and only if $A$ can not be extended, that is: $A$ is maximal if and only if its final configuration is stable); $st(G, g)$ the set of relaxations of $g$.

Furthermore, given $G$, $g$ and

$$A = g \to g_1 \to ... \to g_n$$

an avalanche, the *score vector* of $A$, which we denote $SC_A$, is equal to $(t_v)_{v \in V(G)^*}$, where for any $v \in V(G)^*$ the entry $t_v$ is equal to the number of times node $v$ was fired during the occurrence of $A$.

**Teorema 20** *(The fundamental theorem of sandpiles)*
*Let $G$ be a sandpile graph and let $g$ be a configuration, we have.*

1. *Any avalanche beginning in $g$ is finite.*
2. $|st(G, g)| = 1$.
3. *Given $A$,$B \in Aval_M(G, g)$, we have that $SC_A = SC_B$.*

**Proof.**

- (proof of item 1) We can prove something stronger, we can prove that given $G$ a sandpile graph and given $f$ a configuration on $G$, the length of the maximal avalanches triggered by $f$ is upperbounded by $|V(G)| \, \|f\| \, d(G)$, where $d(G)$ denotes the diameter of $G$ and $\|f\|$ denotes the quantity $\sum_{v \in V(G)^*} f(v)$. We will use the symbol $L(f)$ to denote the length of the maximal avalanches triggered by $f$ (item 3 implies that all the maximal avalanches triggered by $f$ have the same length), the inequality

$$L(f) \leq |V(G)| \, \|f\| \, d(G)$$

Is known as Tardos' bound.

Let $v$ be a node and let $t \leq L(f)$ be a positive integer, we use the symbol $s(v,t)$ to denote the number of firings occurred at $v$ up to time $t$.

**Claim.** Given $v, w$ two nodes and given $t \leq L(f)$ we have that

$$|s(v,t) - s(w,t)| \leq \|f\|$$

(proof of the claim) Let $v, w$ be two nodes such that $s(v,t) \lneqq s(w,t)$. We define $A$ as the set $\{u : s(w,t) \leq s(v,t)\}$ and we define $B$ as the set $\{u : s(v,t) \lneqq s(u,t)\}$. We observe that up to time $t$, all the nodes of $B$ have been fired more than all the nodes of $A$. Also, we have that the number of chips on $B$ has been increased, the total increase is equal to

$$I = \sum_{u \in A, r \in B : \{u,r\} \in E(G)} (s(r,t) - s(u,t))$$

we observe that $I \leq \|f\|$, and it implies that for all $u \in A$ and for all $r \in B$ the inequality $s(u,t) - s(r,t) \leq \|f\|$ holds. Thus, we have that for any pair $x, y \in V(G)^*$ and for any $t \leq L(f)$

$$|s(x,t) - s(y,t)| \leq \|f\|$$

Given $u \in V(G)$ we use the symbol $d(u,s)$ to denote the distance from $u$ to the sink. We have that $s$ doesn't fire, the claim above implies that if $d(v,s) = 1$, then $v$ can fire a most $\|f\|$ times. We can inductively prove that if $d(v,s) = k$, then $v$ can fire at most $k\|f\|$ times. Thus we have that any node of $G$ can fire at most $d(G)\|f\|$ times. Therefore, we have that

$$L(f) \leq |V(G)| \, \|f\| \, d(G)$$

- (proof of item 2) Given $G$ a sandpile graph and given $v$ a node we define the *first order toppling operator* $T_v$ as follows: given $g$ a configuration on $G$ we have that $T_v(f) = f_v$. We observe that the equation

$$T_v(T_u(f)) = T_u(T_v(f))$$

holds, for all $v$, $u$ and $f$. Let $\mathcal{C}(G)$ be the infinite digraph whose vertex set is the set of configurations on $G$ and whose accessibility relation is the relation given by

$$f \rightarrow g \text{ if and only if } \exists v \in V(G)\,(g = T_v(f))$$

We observe that the set of maximal avalanches triggered by $f$ corresponds to the set of maximal $\mathcal{C}(G)$-paths beginning in $f$. Furthermore, we observe that $\mathcal{C}(G)$ has the *confluence property*, that is: given $f, g, h \in V(\mathcal{C}(G))$ if $f \rightarrow g$ and $f \rightarrow h$, there exists $t$ such that $g \rightarrow t$ and $h \rightarrow t$. We have that any digraph satisfying the confluence property holds the following property: any pair of maximal paths beginning at the same node have the same final node [To]. Thus, we have that any pair of maximal avalanches beginning at the same configuration have the same final configuration.

- (proof of item 3) Given $G$ a sandpile graph, we suppose that $V(G) = \{1, ..., n, n+1\}$, and we suppose that $n+1$ is the sink of $G$, the *reduced laplacian* of $G$ is the matrix $L(G) = [a_{ij}]_{i,j \leq n}$ defined by

$$a_{ij} = \begin{cases} -\deg(i), \text{ if } i = j \\ a_{ij} = 1, \text{ if } i \neq j \text{ and } \{i,j\} \in E(G) \\ 0, \text{ otherwise} \end{cases}$$

Suppose that $g$ is a configuration on $G$, we can think of $g$ as an element of $\mathbb{N}^n$. If node $v$ fires, we obtain a new configuration $g_v$. Note that $g_v = g + L_v(G)$, where $L_v(G)$ is the $v$-*th* row of $L(G)$. Thus, we have that for any configuration $g$ and for any maximal avalanche $A$ triggered by $g$ the equality

$$st_G(g) = g + (L(G))^T (SC_A)$$

holds, where $(L(G))^T$ is the transposition of $L(G)$. We call last equation *the motion equation of sandpiles.* This equation has many important consequences. Given $A$ one of the maximal avalanches triggered by $g$, the vector $SC_A$ is a solution of the system

$$st_G(g) - g = (L(G))^T X \textit{ [Sys. 1]}$$

*Kirchhoff's Matrix Theorem* [To] says that $|\det(L(G))|$ is equal to the number of spanning trees of $G$. We note that this quantity is not zero, since $G$ is connected, also $L(G)$ is nonsingular. Then, we have that given $A$ and $B$ two maximal avalanches $SC_A$ and $SC_B$ are equal to the unique solution of the system *[Sys. 1].*

■

Theorem 20 says many things about sandpile automata. Item 1 says that sandpile automata are *terminating*. Item 2 says that sandpile automata are *confluent*, i.e. the input (the initial configuration) determines an unique output (its stabilization). Item 3 says that, though there are many computation paths, sandpile automata are strongly deterministic, since given $\mathcal{SP}(G)$ a sandpile automaton and given two computation paths of $\mathcal{SP}(G)$ on input $g$, the second path is simply a permutation of the first, and as a consequence they have the same length.

Given $\mathcal{C}(G) = \mathbb{N}^{V(G)^*}$ the set of all the configurations on $G$ and given $st(G)$ the set of all the stable configurations on $G$, we can define two functions $st_G : \mathcal{C}(G) \to st(G)$ and $SC_G : \mathcal{C}(G) \to \mathcal{C}(G)$ in the following way:

1. $st_G(g) :=$ the stabilization of $g$.
2. $SC_G(g) := SC_A$, where $A$ is any element of $Aval_M(G, g)$.

Note that, for any sandpile graph $G$, the functions $st_G$ and $SC_G$ are computable, since the avalanches are always finite. given $g$ a configuration on $G$, if one wants to compute either $st_G(g)$ or $SC_G(g)$, one only has to simulate the computation of the automaton $\mathcal{SP}(G)$ on input $g$.

**Notation 21** *Given $g \in \mathcal{C}(G)$ we will use the symbol $SC_g$ to denote the vector $SC_G(g)$.*

Next theorem follows easily from the invariance of the score vector.

**Teorema 22** *Given $G$ a sandpile graph and given $f_1, f_2$ and $f_3$ three configurations, we have that*

1. $st_G(f_1 + f_2 + f_3) = st_G(st_G(f_1 + f_2) + f_3)$.
2. $st_G(f_1 + f_2) = st_G(st_G(f_2) + st_G(f_1))$.

***Proof.*** *It follows easily from the invariance of the score vector* ■

Last theorem allow us to associate to any sandpile graph a sandpile monoid. To this end we define a binary operation $\oplus : st(G)^2 \to st(G)$ in the following way

$$f \oplus g = st_G(f + g)$$

The pair $(st(G), \oplus)$ is a finite commutative monoid. We will use the name *Sandpile Monoid of* $G$ to denote the pair $\mathcal{M}(G) = (st(G), \oplus)$. It is known that the *kernel,* (that is: the intersection of all the ideals), of a finite commutative monoid is an abelian group (see reference [To]). We use the symbol $\mathcal{K}(G)$ to denote the abelian group

$$\left(Ker(\mathcal{M}(G)), \oplus \restriction_{(Ker(\mathcal{M}(G)))^2}\right)$$

which we call *the critical group* (or the sandpile group) of $G$.

**Remark 23** *Given $f, g \in \mathcal{C}(G)$ we can define $f \oplus g$ either as $st_G(f+g)$ or as $st_G(st_G(f) + st_G(g))$. Note that*

$$st_G(f+g) = st_G(f) \oplus st_G(g)$$

The elements of $\mathcal{K}(G)$ are called critical (recurrent) configurations, which encode the long term behavior of the sandpile dynamics on $G$ [BG]. We finish this chapter with Dhar's theorem which can be used to characterize (and recognize) the set of critical configurations. We begin by stating a theorem which characterizes the set of critical configurations, but which is not very useful from the algorithmic point of view.

**Teorema 24** *Given $G$ a sandpile graph and given $f$ a stable configuration on $G$, we have that $f$ is critical if and only if there exists a configuration $g \neq 0$ such that $f \oplus g = f$.*

A proof of this theorem can be found in [To].

**Notation 25** *Given $G$ a sandpile graph, we use the symbol $\delta(G)$ to denote the set*

$$\left\{w \in V(G)^* : \{w, s\} \in E(G)\right\}$$

*We use the symbol $e_{\delta(G)}$ to denote the configuration defined by*

$$e_{\delta(G)}(v) = \textit{number of edges connecting } v \textit{ with } s$$

**Lemma 26** *Given $f$ a stable configuration on $G$ and given $v \in V(G)^*$ we have that*

$$SC_{f+e_{\delta(G)}}(v) \leq 1$$

**Proof.** Let $A$ be a maximal avalanche triggered by $f + e_{\delta(G)}$. We can think of $A$ as a sequence $v_1, ..., v_{L\left(f+e_{\delta(G)}\right)}$ of nodes such that, node $v_i$ is the node fired at time $i$ during the occurrence of $A$. Let $i \leq L\left(f + e_{\delta(G)}\right)$, we use the symbol $f^{(i)}$ to denote the configuration obtained after the *ith* toppling (we suppose that we have chosen one of the maximal avalanches triggered by $f + e_{\delta(G)}$). We will prove, using induction on $i$, that for any $i \leq L\left(f + e_{\delta(G)}\right)$ the following two facts hold:

1. For any $j, k \leq i$, if $j \neq k$ then $v_j \neq v_k$.
2. For all $j \leq i$, the inequality $f^{(i)}(v_j) \lneqq \deg(v_j)$ holds.

- $(i = 1)$ We only have to check item 2. We observe that

$$\begin{aligned} f^{(1)}(v) &= f(v) + e_{\delta(G)}(v) - \deg(v) \\ \text{and } e_{\delta(G)}(v) &\leq \deg(v) \end{aligned}$$

Then, we have that $f^{(1)}(v) \leq f(v) \leq \deg(v)$, since $f$ is stable.

- (Inductive hypothesis) For all $j, k \leq i$ if $j \neq k$, then $v_j \neq v_k$. Furthermore, given $j \leq i$ the inequality $f^{(i)}(v_j) \lneqq \deg(v_j)$ holds.

- $(i+1)$ Let $j, k \leq i+1$, if $j, k \leq i$ and $j \neq k$ then $v_j \neq v_k$. Suppose that $k \lneqq j = i+1$. Then

$$f^{(i+1)}(v_{i+1}) = f(v_{i+1}) - 2\deg(v_{i+1}) + T_{i+1}$$

where $T_{i+1}$ is equal to the number of times the neighbors of $v_{i+1}$ have been fired. The inductive hypothesis implies that no node was fired more than once before time $i$. Also, all the neighbors of $v_{i+1}$ have been fired at most once. It implies that $T_i \leq \deg(v_{i+1})$ and it implies that

$$f^{(i+1)}(v_{i+1}) \leq f(v_{i+1}) - \deg(v_{i+1}) \lneqq 0$$

Now, we pick $l \leq i+1$. We have that

$$f^{(i+1)}(v_l) = f(v_l) - \deg(v_l) + E_{l,i+1}$$

where $E_{l,i+1}$ is equal to the number of times the neighbors of $v_l$ have been fired before time $i+1$. We already know that any node has been fired at most once, also $\deg(v_l) \geq E_{l,i+1}$ and it implies that

$$f^{(i+1)}(v_l) \leq f(v_l) \leq \deg(v_l)$$

■

**Teorema 27** *(Dhar's Theorem) Let $G$ be a sandpile graph and let $f$ be a stable configuration, we have*

1. *$f$ is critical if and only if $f \oplus e_{\delta(G)} = f$.*
2. *$f$ is critical if and only if for any $v \in V(G)^*$ we have that $SC_{f+e_{\delta(G)}}(v) = 1$.*
3. *$f$ is critical if and only if there not exists $A \subseteq V(G)^*$ such that for any $v \in A$ the inequality $f(v) \lneqq \deg_A(v)$ holds, where $\deg_A(v)$ is equal to the number of edges connecting $v$ with a node in $A$.*

**Proof.**

- (item 1). If $f \oplus e_{\delta(G)} = f$, we have that $f$ is critical since $e_{\delta(G)}$ is not null. Given $v \in V(G)^*$ we define the *second order topplig operator* $G_v$ as follows: given $f \in \mathcal{K}(G)$ we have that $G_v(f) = f \oplus e_v$, where $e_v$ is the configuration

$$e_v(w) = \begin{cases} 1 \text{ if } w = v \\ 0 \text{ otherwise} \end{cases}$$

We observe that given $v, w \in V(G)^*$ the equation $G_vG_w = G_wG_v$ holds. We define $T = \prod_{v \in V(G)^*} G_v^{\deg(v)}$ and $H = \prod_{v \in V(G)^*} G_v^{n_v}$, where $n_v = \deg(v) - r_v$, and $r_v$ is equal to the number of edges connecting $v$ with the sink. Note that $id = TH = \prod_{v \in V(G)^*} G_v^{\deg(v)-n_v}$. Also, $\prod_{v \in V(G)^*} G_v^{r_v} = id$. We observe that given $f \in \mathcal{K}(G)$ the equality

$$\left( \prod_{v \in V(G)^*} G_v^{r_v} \right)(f) = f \oplus e_{\delta(G)}$$

holds. Thus, we have that $f = id(f) = f \oplus e_{\delta(G)}$.

- (item 2) If $SC_{f+e_{\delta(G)}}(v) = 1$ for all $v \in V(G)^*$ , then given $v$ a node of $G$ we have that the number of chips that $v$ gets from its neighbors is equal to the number of chips $v$ gives to its neighbors, (we can think that each one of the nodes located at distance 1 from the sink get from $s$ as many nodes as the number of edges connecting them with the sink, since at the beginning we add the configuration $e_{\delta(G)}$). Also, all the nodes of $G$ end with the same amount of chips they had at the beginning of the process (which process? Add $e_{\delta(G)}$ and then relax). Thus, we have that $f \oplus e_{\delta(G)} = f$ and it implies that $f$ is critical since $e_{\delta(G)}$ is not null.

  On the other hand if $f$ is critical we have that $f \oplus e_{\delta(G)} = f$, moreover we know that for any $v$ the inequality $SC_{f+e_{\delta(G)}}(v) \leq 1$ holds. Suppose that there exists $v$ such that $SC_{f+e_{\delta(G)}}(v) = 0$. Then, there exists $v$ such that $SC_{f+e_{\delta(G)}}(v) = 0$ and $SC_{f+e_{\delta(G)}}(w) = 1$ for some $w$ neighbor of $v$. It implies that $\left(f \oplus e_{\delta(G)}\right)(v) \gneqq f(v)$, but it is clearly a contradiction.

- (item 3) Suppose that $f$ is not critical, then the set $A$ defined by

$$A = \left\{u : SC_{f+e_{\delta(G)}}(v) = 0\right\}$$

  is nonempty. Suppose that there exists $v \in A$ such that $f(v) \geq \deg_A(v)$. We have that $\deg(v) - \deg_A(v)$ neighbors are fired during the relaxation of $f + e_{\delta(G)}$. Also, when the relaxation process comes to an end, there are $f(v) + \deg(v) - \deg_A(v)$ chips at $v$. It is clear that

$$f(v) + \deg(v) - \deg_A(v) \geq \deg(v)$$

  Then, we have that node $v$ becomes unstable, but it is impossible since after the relaxation any node of $v$ become stable.

Now, we will suppose that there exists $A$ such that for all $v \in A$ the inequality $f(v) \lneqq \deg_A(v)$ holds. We will prove that for all $v \in A$ it follows that $SC_{f+e_{\delta(G)}}(v) = 0$.

Let $v$ be a node in $A$ and let $t \leq L\left(f + e_{\delta(G)}\right)$ be a positive integer. We use induction on $t$ to prove that for all $t \leq L\left(f + e_{\delta(G)}\right)$ and for all $v \in A$, node $v$ is not fired before time $t+1$.

- $(t = 0)$ Node $v$ can not be fired before $t = 1$, since $f(v) \lneqq \deg_A(v)$.
- $(i \leq t)$ We suppose the claim true for all $i \leq t$.
- $(t+1)$ Let $B = V(G)^* - A$. The inductive hypothesis implies that
$$f^{(t)}(v) \leq f(v) + \deg_B(v)$$
Then
$$f^{(t)}(v) \lneqq \deg_A(v) + \deg_B(v) = \deg(v)$$
Thus, we have that at time $t$ node $v$ is stable, then it can not be fired at time $t+1$.

■

**Corollary 28** *If $f$ is a critical configuration, then given $v, w$ two nodes such that $\{v, w\} \in E(G)$ we have that either $f(v) \neq 0$ or $f(w) \neq 0$.*

**Proof.** This fact is a consequence of the third item of the theorem above. Suppose that there exist $v, w \in V(G)^*$ such that $\{x, y\} \in E(G)$ and $f(v) = f(w) = 0$. Let $A = \{x, y\}$, we have that $f(x), f(y) \lneqq 1 = \deg_A(x) = \deg_A(y)$ ■

We can use item 1 as the basis of a linear time (real time with respect to the size of $G$) algorithm which recognizes the set of critical configurations, this algorithm is called *The Burning Test* ($BT$ for short) and works as follows:

On input $(G, f)$, (where $G$ is a sandpile graph and $f$ is an stable configuration on $G$), algorithm $BT$ performs the computation described below do

1. $BT$ computes $f + e_{\delta(G)}$.
2. $BT$ simulates the relaxation process of $f + e_{\delta(G)}$.
3. $BT$ counts the number of times each node was fired during the relaxation.
4. If all the nodes were fired exactly once $BT$ accepts, otherwise $BT$ rejects.

We can use the burning test as well, to prove that there exists a bijection between the set of critical configurations on $G$ and the set of spanning trees of $G$, that is: we can use the burning test to prove that $|\mathcal{K}(G)| = |\det(L(G))|$ [Ba].

## 3.1 Exercises

1. Prove theorem 24.
2. Prove that $|\mathcal{K}(G)| = |\det(L(G))|$.
3. Define a directed version of The Abelian Sandpile Model, and check which of the basic properties of The Abelian Sandpile Model hold in the directed case.

# 4 Algorithmic problems

In this chapter we introduce the algorithmic problems (predicting tasks) that we want to analyze. Moreover, we study the relative hardness of those problems.

## 4.1 The algorithmic problems

The Sandpile Prediction Problem, is the algorithmic problem defined by:

**Problem 29** *(SPP, sandpile prediction)*

- *Input:* $(G, g)$*, where* $G$ *is a sandpile graph and* $g \in \mathcal{C}(G)$.
- *Problem: Compute* $st_G(g)$.

**Remark 30** *Tardos' bound [?] implies that* $SPP$*, and each one of the algorithmic problems introduced below, can be solved in polynomial time, because of this we will analyze the relative complexity of those problems using the notion of* $NC$*-Turing reducibility.*

A Second problem is $MC$, which corresponds to the computation of the monoid operation $\oplus$.

**Problem 31** *(MC, monoid computations)*

- *Input:* $(G, f, g)$*, where* $G$ *is a sandpile graph and* $f, g \in \mathcal{M}(G)$.

- *Problem: Compute* $f \oplus g$.

We know of the existence of a special set of configurations: the set of critical configurations. Can we recognize critical configurations?

**Problem 32** *(RR, recognition of critical configurations)*

- *Input:* $(G, f)$, *where* $G$ *is a sandpile graph and* $f \in \mathcal{C}(G)$.
- *Problem: Decide if* $f$ *belongs to* $\mathcal{K}(G)$.

Now, we introduce the problem $GC$ which is the restriction of $SPP$ to critical configurations.

**Problem 33** *(GC, group computations)*

- *Input:* $(G, f, g)$, *where* $G$ *is a sandpile graph and* $f, g \in \mathcal{K}(G)$.
- *Problem: Compute* $f \oplus g$.

**Problem 34** *($MC^*$, mixed computations)*

- *Input:* $(G, f, g)$, *where* $G$ *is a sandpile graph,* $f \in \mathcal{K}(G)$ *and* $g \in \mathcal{M}(G)$.
- *Problem: Compute* $f \oplus g$.

**Problem 35** *($CSV$, computation of score vectors)*

- *Input:* $(G, f)$, *where* $G$ *is a sandpile graph and* $f \in \mathcal{C}(G)$.
- *Problem: Compute the vector* $SC_f$.

It is clear that the problem $CSV$ is equivalent to the counting problem consisting in computing the number of times an input-node is toppled during the relaxation of a input-configuration. We introduce a related problem which seems to be easier than $CSV$.

**Problem 36** *($SPA$, Sandpile Accessibility)*

- *Input:* $(G, f, v)$, *where* $G$ *is a sandpile graph,* $f \in \mathcal{C}(G)$ *and* $v \in V(G)^*$
- *Problem: decide if* $SC_f(v) \gneqq 0$.

Given $G$ a sandpile graph, we use the symbol $e_{\mathcal{K}(G)}$ to denote the identity of $\mathcal{K}(G)$.

**Problem 37** *(IC, computation of identities)*

- *Input:* $G$, *where* $G$ *is a sandpile graph.*

- *Problem: Compute* $e_{\mathcal{K}(G)}$.

Remember that given $V \in (G)^*$, we use the symbol $e_v$ to denote the configuration

$$e_v(w) = \begin{cases} 1 \text{ if } v = w \\ 0, \text{ otherwise} \end{cases}$$

Let $e_G : V(G)^* \to \mathcal{K}(G)$ be the function defined by $e_G(v) = e_{\mathcal{K}(G)} \oplus e_v$

**Problem 38** *(EC, computation of e)*

- *Input:* $(G, v)$, *where* $G$ *is a sandpile graph and* $v \in V(G)^*$.
- *Problem: Compute* $e_G(v)$.

**Remark 39** *(measuring the size of the instances)*

1. *Given* $(G, f)$ *an instance of either SPP or CSV we measure its size as* $|G| + \|f\|$.
2. *Given* $(G, f, g)$ *an instance of either MC,* $MC^*$ *or GC we measure its size as* $|G|$.
3. *Given* $(G, f)$ *an instance of RR we measure its size as* $|G|$.
4. *Given* $(G, f, v)$ *an instance of SPA, we measure its size as* $|G| + \|f\|$.

## 4.2 The relative hardness of sandpile prediction problems

Given $\mathcal{C}$ a class of sandpile graphs and given $\mathcal{L}$ one of the algorithmic problems defined above, we will use the symbol $\mathcal{L}[\mathcal{C}]$ to denote the restriction of $\mathcal{L}$ to the class $\mathcal{C}$. We are mainly interested in classes of low dimensional sandpile lattices. We will begin our work from a very general point of view: we will consider a large family of well-behaved sandpile classes, which we call bounded classes.

**Definition 40** *A bounded class is a class* $\mathcal{C}$ *of sandpile graphs such that*

1. *There exists* $D_{\mathcal{C}} \geq 2$ *such that for any* $G \in \mathcal{C}$ *and for all* $v \in V(G)^*$ *we have that* $2 \leq \deg(v) \leq D_{\mathcal{C}}$.
2. *Given* $G \in \mathcal{C}$, *there not exists a pair* $(A, v)$ *such that* $A \subset V(G)^*$, $v \in A$, $|A| \geq 2$ *and for all* $w \in A$ *we have that any path connecting* $w$ *with the sink of* $G$ *visits the node* $v$, *(that is: the elements of* $\mathcal{C}$ *are free of bottlenecks).*

**Teorema 41** *Let* $\mathcal{C}$ *be a bounded class*

1. *$SPP[\mathcal{C}]$ and $CSV[\mathcal{C}]$ are $NC^2$-Turing equivalent.*

2. *$SPP[\mathcal{C}]$ and $MC[\mathcal{C}]$ are $NC^1$-Turing equivalent.*

3. *$MC[\mathcal{C}]$ is $NC^2$ reducible to $MC^*[\mathcal{C}]$.*

4. *$MC^*[\mathcal{C}]$ can be solved in time $O(\log^2 n)$ using a polynomial number of processors, if oracle access to $GC[\mathcal{C}]$and $EC[\mathcal{C}]$ is provided.*

5. *$EC[\mathcal{C}]$ is $NC^2$ reducible to $GC[\mathcal{C}]$.*

**Proof.**

- (proof of item 1) It follows easily from the proof of the third item of theorem 20.

- ( proof of item 2) It is clear that $MC[\mathcal{C}]$ is $NC^1$-Turing reducible to $SPP[\mathcal{C}]$, also we prove that $SPP[\mathcal{C}]$ is $NC^1$-Turing reducible to $MC[\mathcal{C}]$. We can suppose, without loss of generality, that there exists $m$ such that $\|f\| = 2^m$. We can express the configuration $f$ as a sum $\sum\limits_{i \leq 2^m} f_i$ such that for any $i \leq 2^m$ we have $\|f_i\| \leq 1$ (hence, all the configurations $f_i$ are stable). Theorem 22 implies that

$$st_G(f) = st_G\left(\sum_{i \leq 2^m} f_i\right) = st_G\left(\sum_{i \leq 2^{m-1}} (f_{2i-1} \oplus f_{2i})\right)$$

  So, instead of computing $st_G(f)$, we compute $\{f_{2i-1} \oplus f_{2i} : i \leq 2^{m-1}\}$, and then we compute $st_G\left(\sum\limits_{i \leq 2^{m-1}} (f_{2i-1} \oplus f_{2i})\right)$. Furthermore, we can iterate this procedure reducing to the half the number of summands in each iteration. If we perform $m$ iterations, we obtain a pair $g_1, g_2 \in \mathcal{M}(G)$ such that $st_G(f) = g_1 \oplus g_2$. Then, we can compute $st_G(f)$ in time $O(\log(\|f\|))$, using $O(\|f\|)$ processors and asking at most $\log(\|f\|)$ queries to the $MC$-oracle

- (proof of item 3) Let $G \in \mathcal{C}$ and let $f, g$ be two stable configurations, we show that we can compute the score vector of the maximal avalanches triggered by $f + g$ in time $O\left(\log^2(|V(G)| + \|f\| + \|g\|)\right)$, if oracle access to $MC^*[\mathcal{C}]$ is provided. Given $f$ a stable configuration we use the symbol $f^*$ to denote the configuration $w_G - f$. We observe that:

1. $SC(f+g, f^*) = SC(f,g) + SC(f \oplus g, f^*)$.
2. $SC(f+g, g^*) = SC(f,g) + SC(f \oplus g, g^*)$.

3. $SC(3(f+g)+w_G, f^*+g^*) = 3SC(f,g) + SC(f\oplus g, f^*) + SC(f\oplus g, g^*) + SC(w_n \oplus g, w_n \oplus f) + SC(w_n, 2w_n \oplus f \oplus g) + SC(3w_n \oplus f \oplus g, f \oplus g)$.

4. We can compute $SC(f+g, f^*)$ since $st_G(f+g+f^*) = w_G \oplus g$ and we have oracle access to $MC^*[\mathcal{C}]$.

5. We can compute $SC(f+g, g^*)$ since $st_G(f+g+g^*) = w_G \oplus f$.

6. We can compute $SC(3(f+g)+w_G, f^*+g^*)$ since

$$st_G(3(f+g)+w_G+f^*+g^*) = w_G \oplus (3(f+g)+f^*+g^*)$$

and we can also compute $SC(3w_G \oplus f \oplus g, f \oplus g)$ since

$$w_G \oplus f \oplus g \oplus f \oplus g = w_G \oplus (2(w_G+f+g))$$

Thus we can compute three vectors $X, Y$ and $Z$ such that

$$\begin{aligned} X &= SC(f,g) + SC(f\oplus g, f^*) \\ Y &= SC(f,g) + SC(f\oplus g, g^*) \\ Z &= 3SC(f,g) + SC(f\oplus g, f^*) + SC(f\oplus g, g^*) \end{aligned}$$

Note that the matrix

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 3 & 1 & 1 \end{bmatrix}$$

is nonsingular. It implies that we can compute $SC(f,g)$ and $st_G(f+g) = f \oplus g$ in polylogarithmic time. If we suppose that we have oracle access to $MC^*[\mathcal{C}]$, we can run the algorithm defined above in time $O\left(\log^2(|V(G)| + \|f\| + \|g\|)\right)$ using a polynomial number of processors

- (proof of item 4) Let $G \in \mathcal{C}$, let $v \in V(G)^*$ and let $w_v = w_G - e_v$. We claim that $w_v$ is a critical configuration. It is easy to check that there not exists $A \subseteq V(G)^*$ such that for any $u \in A$ we have that $\deg_A(u) \gneqq w_v(u)$. Theorem **??** implies that $w_v$ is a critical configuration. Now, we observe that

$$\begin{aligned} e_G(v) &= e_v \oplus e_{\mathcal{K}(G)} = e_v \oplus \left(w_v \oplus w_v^{-1}\right) \\ &= (e_v \oplus w_v) \oplus w_v^{-1} = w_G \oplus w_v^{-1} \end{aligned}$$

Thus, if one wants to compute $e_G(v)$, it is sufficient to compute $w_G \oplus w_v^{-1}$. Note that $w_G, w_v^{-1} \in \mathcal{K}(G)$.

We can compute $w_v^{-1}$ in time $O\left(\log^2(|V(G)| + \|f\| + \|g\|)\right)$ if oracle access to $GC[\mathcal{C}]$ is provided because $w_v^{-1} = (w_v)^{|\mathcal{K}(G)|-1}$ and it is known that $|\mathcal{K}(G)|$ is equal to $|\det(L^s(G))| \leq D_{\mathcal{C}}^{|V(G)|}$.

If we put all the pieces together we can define an algorithm $\mathcal{N}$, which has oracle access to $GC[\mathcal{C}]$, and which compute the function $e_{\mathcal{C}}$ in time $O\left(\log^2\left(|V(G)| + \|f\| + \|g\|\right)\right)$ using $O\left(|V(G)|^2\right)$ processors.

On input $(G, v)$ algorithm $\mathcal{N}$ works in the following way:

1. $\mathcal{N}$ computes $|\det(L^s(G))|$, in time $O\left(\log^2\left(|V(G)|\right)\right)$ using $O\left(|V(G)|^2\right)$ processors.

2. Using fast exponentiation and a $GC[\mathcal{C}]$-oracle, algorithm $\mathcal{N}$ computes $(w_v)^{|\det(L^s(G))|-1}$ in time $O(\log(|V(G)|))$.

3. $\mathcal{N}$ computes $w_G \oplus w_v^{-1}$.

4. $\mathcal{N}$ prints $w_G \oplus w_v^{-1}$.

- (proof of item 5) Let $(G, f, g)$ be an instance of $MC^*[\mathcal{C}]$. We observe that
$$f \oplus g = f \oplus g \oplus \underbrace{e_{\mathcal{K}(G)} \oplus ... \oplus e_{\mathcal{K}(G)}}_{\|g\|\text{-times}}$$
If we express $g$ as $\sum_{v \in V(G)^*} m_v e_v$ we get
$$f \oplus g = f \oplus \left( \bigoplus_{v \in V(G)^*} m_v e_G(v) \right)$$
Also, we can use $\left|V(G)^*\right|$ processors to compute $\{m_v e_G(v)\}_{v \in V(G)^*}$, this computation takes $O\left(\log^2\left(|V(G)| + \|f\|\right)\right)$ time units, since we are supposing that we have oracle access to $GC[\mathcal{C}]$. We can use the same $\left|V(G)^*\right|$ processors to compute $f \oplus \left( \bigoplus_{v \in V(G)^*} m_v e_G(v) \right)$ in time $O(\log(|V(G)| + \|f\|))$. Thus, we have proven that $MC^*[\mathcal{C}]$ is $NC^2$-Turing reducible to $GC[\mathcal{C}]$

■

**Corollary 42** *Let $\mathcal{C}$ be a bounded class of sandpile graphs, the problems $SPP[\mathcal{C}]$, $CSV[\mathcal{C}]$, $MC[\mathcal{C}]$, $MC^*[\mathcal{C}]$, $EC[\mathcal{C}]$ and $GC[\mathcal{C}]$ are $NC^2$-Turing equivalent.*

**Proof.** It is clear that $GC[\mathcal{C}]$ is $NC^2$-Turing reducible to $SPP[\mathcal{C}]$, since $GC[\mathcal{C}]$ is a restriction of $SPP[\mathcal{C}]$ ■

**Proposition 43** *(The hardness of the easiest problems)*

1. *$IC[\mathcal{C}]$ is $NC^2$ reducible to $GC[\mathcal{C}]$.*
2. *$RR[\mathcal{C}]$ is $NC^1$ reducible to $SPA[\mathcal{C}]$.*
3. *$SPA[\mathcal{C}]$ is $NC^1$ reducible to $CSV[\mathcal{C}]$.*

**Proof.**

- (proof of item 1) It is clear that $e_{\mathcal{K}(G)} = (w_G)^{|\mathcal{K}(G)|} = (w_G)^{|\det(L_r^s(G))|}$. Also, we can compute $e_{\mathcal{K}(G)}$ in time $O\left(\log^2(|G|)\right)$ if oracle access to $GC[\mathcal{C}]$ is provided.
- (proof of item 2) We observe that $(G, f) \in RR[\mathcal{C}]$ if and only if $f$ is stable and for any $v \in V(G)^*$ we have that $SC_{f+e_{\delta(G)}}(v) \gneqq 0$. We can check these two properties in time $O(\log(|G|))$ if oracle access to $SPA[\mathcal{C}]$ is provided.
- (proof of item 3) It is clear that $SPA[\mathcal{C}]$ is $NC^2$-Turing reducible to $CSV[\mathcal{C}]$.

■

**Corollary 44** *Let $\mathcal{C}$ be a bounded class of sandpile graphs, the problems $SPP[\mathcal{C}]$, $CSV[\mathcal{C}]$, $MC[\mathcal{C}]$, $MC^*[\mathcal{C}]$, $EC[\mathcal{C}]$, $GC[\mathcal{C}]$, $I[\mathcal{C}]$, $SPA[\mathcal{C}]$ and $RR[\mathcal{C}]$ are ptime computable.*

**Proof.** Tardos' bound implies that $SPP[\mathcal{C}]$ is *ptime* computable ■

## 4.3 Exercises

1. Is it the Abelian Sandpile Model a class of Boltzmann Systems? Can be the Abelian Sandpile Model turned into a class of Boltzmann Systems?
2. Prove that $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_3$ are bounded classes.
3. Prove that given $\mathcal{C}$ a bounded class, given $G \in \mathcal{C}$ and given $v \in V(G)^*$, the configuration $w_v = w_G - e_v$ is a critical one.

# 5
# Statistics of critical avalanches

We will use the term *critical avalanches* to denote the avalanches triggered by the addition of two critical configurations. In this section we study the length of critical avalanches, that is: we establish upper and lower bounds on the possible length of critical avalanches.

Given $f, g \in \mathcal{K}(G)$ we will use the symbol $L(f,g)$ to denote the length of the critical avalanches triggered by $f+g$. Given $f,g$ two configurations on $G$, we use the symbol $f \leq g$ to denote that for all $v \in V(G)^*$ the inequality $f(v) \leq g(v)$ holds. Note that

1. If $f \leq g$ and $h \leq r$, then $L(f,h) \leq L(g,r)$.
2. For any $f,g \in \mathcal{K}(G)$ we have $L(f,g) \leq L(w_G, w_G)$.

From now on we will use the symbol $\mathcal{C}$ to denote an arbitrary bounded class of sandpile graphs. First at all we prove that all the critical avalanches are long, that is: we prove that the length of a critical avalanche can not be sublinear.

**Teorema 45** *(Critical configurations can only generate long avalanches) Given $G \in \mathcal{C}$ and given $f \in \mathcal{K}(G)$ we have*

$$\forall g \in \mathcal{K}(G)\left(L(f,g) \geq \frac{\left|V(G)^*\right| - D_{\mathcal{C}}\left|\delta(G)\right|}{D_{\mathcal{C}}}\right)$$

**Proof.** Let $H(G) = \sum_{v \in V(G)^*} (\deg(v) - 1)$. A configuration $f$ is recurrent if and only if

1. for any $v \in V(G)^*$ we have $SC_{f+e_{\delta(G)}}(v) = 1.$
2. $st_G\left(f + e_{\delta(G)}\right) = f.$

Suppose that we run the avalanche triggered by $f + e_{\delta(G)}$ and for any $v \in V(G)^*$ we count the number of grains on $v$ just before the node $v$ is toppled. Let $\alpha$ be equal to the result of our counting. Note that $\alpha \geq H(G) + \left|V(G)^*\right|$. On the other hand, it is easy to verify that we count twice the grains which remain on $V(G)^*$ after the avalanche, and we count once the lost grains. So we have

$$2\|f\| + D_{\mathcal{C}}|\delta(G)| \geq H(G) + \left|V(G)^*\right|$$

Thus, we have that

$$\|f\| \geq \left(\frac{H(G) + \left(\left|V(G)^*\right| - D_{\mathcal{C}}|\delta(G)|\right)}{2}\right)$$

Now, given $f, g \in \mathcal{K}(G)$ we have that

$$\|f\| + \|g\| \geq H(G) + \left(\left|V(G)^*\right| - D_{\mathcal{C}}|\delta(G)|\right)$$

and it implies that, when we begin with the configuration $f + g$, we have to throw at least $\left(\left|V(G)^*\right| - D_{\mathcal{C}}|\delta(G)|\right)$ grains. We can throw at most $D_{\mathcal{C}}$ grains per toppling, and it implies that

$$L(f,g) \geq \frac{\left|V(G)^*\right| - D_{\mathcal{C}}|\delta(G)|}{D_{\mathcal{C}}}$$

■

**Corollary 46** *Suppose that for any $G \in \mathcal{C}$ we have that $K \geq |\delta(G)|$, then for any $f, g \in \mathcal{K}(G)$*

$$L(f,g) \geq \frac{\left|V(G)^*\right| - D_{\mathcal{C}}K}{D_{\mathcal{C}}} \in \Omega\left(\left|V(G)^*\right|\right)$$

Now, we will establish an upper bound on avalanche length. Let $G$ be an element of $\mathcal{C}$ whose sink is equal to $s$, remember that the symbol $d(G)$ denotes the quantity $\max_{v \in V(G)^*}\{d_G(v,s)\}$, where $d_G(v,s)$ is the distance from $v$ to the sink.

**Teorema 47** $L(w_G, w_G) \in O\left(\left|V(G)^*\right| d(G)^2\right).$

**Proof.** Note that $d(G) \in O(|V(G)|)$. Given $i \leq d(G)$, we use the symbol $N_i(G)$ to denote the set $\{v : d_G(v,s) = i\}$. Note that $V(G)^* =$

$\left(\bigsqcup_{1\leq i\leq d(G)} N_i\left(G\right)\right)$, (where the symbol $\sqcup$ denotes that the union is a disjoint union). If we define $\mu_i$ as the number of topplings occurred on $N_i\left(G\right)$ during the relaxation process, we have that

$$L\left(w_G, w_G\right) = \sum_{i\leq d(G)} \mu_i$$

We observe that for any $i \leq d\left(G\right)$ the inequality

$$\mu_i - \mu_{i+1} \leq 2D_{\mathcal{C}} \left|V\left(G\right)^*\right|$$

holds, (where, by convention, $\mu_{d(G)+1} = 0$). Thus, we have

$$\begin{aligned} L\left(w_G, w_G\right) &= \sum_{i\leq d(G)} \mu_i = \sum_{i\leq d(G)} \left(\sum_{j=i} \mu_j - \mu_{j+1}\right) \\ &\leq \sum_{i\leq d(G)} \left(\sum_{j=i} 2D_{\mathcal{C}} \left|V\left(G\right)^*\right|\right) = 2D_{\mathcal{C}} \left|V\left(G\right)^*\right| \sum_{i\leq d(G)} i \end{aligned}$$

Note that

$$\left(2D_{\mathcal{C}} \left|V\left(G\right)^*\right| \sum_{i\leq d(G)} i\right) \in O\left(\left|V\left(G\right)^*\right| d\left(G\right)^2\right)$$

Thus, we have proven that $L\left(w_G, w_G\right) \in O\left(V\left(G\right)^* d\left(G\right)^2\right)$ ■

**Remark 48** *It is important to remark that the best, already established, upper bound for avalanche length on general graphs is Tardos´s bound (see reference [T]). Tardos´s bound implies that for any sandpile graph G*

$$L\left(w_G, w_G\right) \in O\left(\left|V\left(G\right)^*\right|^3 d\left(G\right)\right)$$

*Observe that* $\left|V\left(G\right)^*\right| d\left(G\right)^2 \lneqq \left|V\left(G\right)^*\right|^3 d\left(G\right)$, *for any graph of size bigger than* 1.

Now, we will establish a lower bound on $L\left(w_G, w_G\right)$ which could be stronger than the linear bound of theorem 45.

**Teorema 49** $L\left(w_G, w_G\right) \in \Omega\left(\left|V\left(G\right)^*\right| + d\left(G\right)^2\right)$.

**Proof.** Remember that all the avalanches triggered by $2w_G$ have the same length. Given $G \in \mathcal{C}$ we will lowerbound the length of a very specific avalanche triggered by $2w_G$. Given $i \leq d(G)$, we define $M_i(G)$ as the induced subgraph of $G$ constituted by the subset of nodes $\bigcup_{j=i}^{d(G)} N_i(G)$, we can think of the graph $M_i(G)$ as a sandpile graph whose sink is equal to $N_{i-1}(G)$ (we define $N_0(G) = \{s\}$). Given $0 \lneqq i \leq d(G) - 1$, we use the symbol $w_i$ to denote the configuration $w_{M_i(G)}$. We have that the subgraph $M_{i+1}(G)$ is embedded into $M_i(G)$. Note that

$$2w_i = \left(w_i + e_{\delta(M_i(G))}\right) + \left(w_{i+1} + \beta_i\right)$$

where $\beta_i$ is some configuration. Theorem 22 says that

$$st_{M_i(G)}(2w_i) = st_{M_i(G)}\left(st_{M_i(G)}\left(w_i + e_{\delta(M_i(G))}\right) + st_{M_i(G)}\left(w_{i+1} + \beta_i\right)\right)$$

and Theorem **??** says that:

1. $st_{M_i(G)}\left(w_i + e_{\delta(M_i(G))}\right) = w_i$.
2. $L\left(w_i, e_{\delta(M_i(G))}\right) = |M_i(G)|$.

Thus, we have that there exists a configuration $\gamma_1$ such that we can pass from the configuration $2w_1$ to the configuration $2w_2 + \gamma_1$. Furthermore, we have that the partial avalanche carrying us from $2w_1$ to $2w_2 + \gamma_1$ has a length equal to $|M_1(G)|$, this partial avalanche (it is not a maximal avalanche) is the first stage of the whole stabilization process. In the second stage we work on the subgraph $M_2(G)$ with the configuration $2w_2$. We can now claim that after $|M_2(G)|$ topplings we can pass from $2w_2$ to $2w_3 + \gamma_3$ for some configuration $\gamma_3$. If we continue in this way, going to $N_{d(G)}(G) = M_{d(G)}(G)$ (the *core* of $G$), we will generate $d(G)$ partial avalanches whose length are lowerbounded by $|M_1(G)|, |M_2(G)|, ..., \left|M_{d(G)}(G)\right|$ (respectively). Therefore, we have that

$$L(w_G, w_G) \geq \sum_{i=1}^{d(G)} |M_i(G)|$$

We observe that:

1. $|M_1(G)| = \left|V(G)^*\right|$.
2. For all $i \lneqq d(G)$ we have that $|M_i(G)| \gneqq |M_{i+1}(G)|$.

Therefore, we have that $L(w_G, w_G) \in \Omega\left(\left|V(G)^*\right| + d(G)^2\right)$ ■

**Corollary 50** *If $\mathcal{C}$ is a bounded class of sandpile graphs such that $d(G) \notin O\left(\sqrt{|V(G)|^*}\right)$, then $L(w_G, w_G) \notin O\left(|V(G)|^*\right)$.*

Let $\mathcal{C}$ be a bounded class. We will prove that if we choose uniformly at random two critical configurations $f$ and $g$, then with high probability the avalanche triggered by $f+g$ is large, its length is almost equal to the length of the longest critical avalanche. First at all we have to remember the notion of accessibility. Given $f,g \in C(G)$ we say that $g$ is *accessible* from $f$ if and only if there exists a configuration $h \geq g$ and there exists a configuration $t$ such that

$$h = f + (L(G))^T(t)$$

We will use the symbol $f \rightarrow g$ to indicate that $g$ is accessible from $f$.

**Notation 51** *Given $G$ a sandpile graph, we use the symbol $\overrightarrow{1}$ to denote the configuration defined by*

$$\textit{for all } v \in V(G)^* \textit{ we have that } \overrightarrow{1}(v) = 1$$

**Lemma 52** *Given $G \in \mathcal{C}$, we have that for any $f_1, ..., f_{2(D_\mathcal{C})^2} \in \mathcal{K}(G)$ the configuration $2w_G$ is accessible from $f_1 + ... + f_{2(D_\mathcal{C})^2}$*

**Proof.** Remember that given $f \in \mathcal{K}(G)$ and given $\{v,w\} \in E(G)$, either $f(w) \gneqq 0$ or $f(v) \gneqq 0$ (theorem **??**). Let $f_1, ..., f_{D_\mathcal{C}+1}$ be $D_\mathcal{C}+1$ critical configurations, given $v \in V(G)^*$ we have that either there exists $i \leq D_\mathcal{C}+1$ such that $f_i(v) \gneqq 0$ or for any $w$ neighbor of $v$ and for any $i \leq D_\mathcal{C}+1$ we have that $f_i(w) \gneqq 0$. Suppose that for all $i \leq D_\mathcal{C}+1$ we have that $f_i(v) = 0$, in this case we can choose any neighbor of $v$, say $w$, and fire it. Also, we can place at least one chip on $v$, taking care of leaving at least 1 chip on $w$. It is clear that, if we begin with the configuration $\sum\limits_{i \leq D_\mathcal{C}+1} f_i$ we can choose a sequence of topplings, of length at most $\left|V(G)^*\right|$, such that if we apply this sequence on $\sum\limits_{i \leq D_\mathcal{C}+1} f_i$, we obtain a configuration $h$ which is different to zero on any $v \in V(G)^*$, that is: there exists $h \geq \overrightarrow{1}$ such that $\sum\limits_{i \leq (D_\mathcal{C})^2} f_i \rightarrow h$. Thus, given $f_1, ..., f_{2(D_\mathcal{C})^2} \in \mathcal{K}(G)$ there exists $h \geq 2w_G$ such that $\sum\limits_{i \leq 2(D_\mathcal{C})^2} f_i \rightarrow h$ ■

**Teorema 53** *(critical configurations generate, with high probability, long avalanches) Given $G \in \mathcal{C}$ we have*

$$\Pr_{f,g \in \mathcal{K}(G)} \left[ L(f,g) \geq \frac{L(w_G, w_G)}{4^{(D_\mathcal{C})^2}} \right] \geq \frac{1}{2(D_\mathcal{C})^2}$$

**Proof.** Let $\alpha = 2\left(D_{\mathcal{C}}\right)^2$, given $f_1, f_2, ..., f_\alpha$ we have that $\sum\limits_{i \leq \alpha_G} f_i \rightarrow 2w_G$. It implies that

$$L\left(f_\alpha, \sum_{i \leq \alpha - 1} f_i\right) \geq L\left(w_G, w_G\right)$$

Also, we have that either

$$L\left(f_\alpha, \bigoplus_{i \leq \alpha - 1} f_i\right) \geq \frac{L\left(w_G, w_G\right)}{2}$$

or

$$L\left(\sum_{i \leq \alpha - 1} f_i\right) \geq \frac{L\left(w_G, w_G\right)}{2}$$

Arguing in this way we can prove that there exists $i \leq \alpha$ such that

$$L\left(f_i, \bigoplus_{j \leq i - 1} f_j\right) \geq \frac{L\left(w_G, w_G\right)}{2^\alpha}$$

Thus, we have that

$$\Pr_{f_1, ..., f_\alpha}\left[\exists i \leq \alpha \left(L\left(f_i, \bigoplus_{j \leq i-1} f_j\right) \geq \frac{L\left(w_G, w_G\right)}{2^\alpha}\right)\right] = 1$$

Note that for any $f \in \mathcal{K}\left(G\right)$ and for any $i \geq 1$

$$\Pr_{f_1, ..., f_i}\left[\bigoplus_{j \leq i} f_j = f\right] = \frac{1}{\left|\mathcal{K}\left(G\right)\right|}$$

i.e. if we choose uniformly at random $i$ critical configurations and we compute their sum, then we generate uniformly at random one element of $\mathcal{K}\left(G\right)$. Given $f_1, ..., f_\alpha$ a sequence of critical configurations on $G$ and given $j \leq \alpha - 1$, we define $g_i = \bigoplus\limits_{j \leq i} f_j$. We have that:

1. The procedure below is a sound method to generate uniformly at random two elements of $\mathcal{K}\left(G\right)$ :
   - Choose uniformly at random $f_1, ..., f_\alpha$.
   - Choose uniformly at random $i \in \{2, ..., \alpha\}$.
   - Compute $f_i$ and $g_{i-1}$.

2. The following equality holds

$$\Pr_{f_1,\dots,f_\alpha}\left[\exists 2 \leq i \leq \alpha \left(L\left(f_i, g_{i-1}\right) \geq \frac{L\left(w_G, w_G\right)}{2^\alpha}\right)\right] = 1$$

From items 1 and 2 we have that

$$\Pr_{2\leq i\leq\alpha}\left[L\left(f_i, g_{i-1}\right) \geq \frac{L\left(w_G, w_G\right)}{2^\alpha}\right] \geq \frac{1}{\alpha - 1}$$

and

$$\begin{aligned}\Pr_{f,g\in\mathcal{K}(G)}\left[L\left(f, g\right) \geq \frac{L\left(w_G, w_G\right)}{2^\alpha}\right] &= \\ \Pr_{2\leq i\leq\alpha;\ f_1,\dots,f_\alpha}\left[L\left(f_i, g_{i-1}\right) \geq \frac{L\left(w_G, w_G\right)}{2^\alpha}\right] &\geq \frac{1}{\alpha - 1}\end{aligned}$$

Thus, we have proven that

$$\Pr_{f,g\in\mathcal{K}(G)}\left[L\left(f, g\right) \geq \frac{L\left(w_G, w_G\right)}{4^{(D_\mathcal{C})^2}}\right] \geq \frac{1}{2\left(D_\mathcal{C}\right)^2}$$

■

Summarizing we have

**Teorema 54** *Let $\mathcal{C}$ be bounded class of sandpile graphs and let $G$ be an element of $\mathcal{C}$.*

1. *For all $f, g \in \mathcal{K}(G)$ we have that $L(f,g) \geq \frac{|V(G)^*| - D_\mathcal{C}|\delta(G)|}{D_\mathcal{C}}$*
2. $\Pr_{f,g\in\mathcal{K}(G)}\left[L(f,g) \geq \frac{L(w_G,w_G)}{4^{(D_\mathcal{C})^2}}\right] \geq \frac{1}{2(D_\mathcal{C})^2}$.
3. $L(w_G, w_G) \in O\left(\left|V(G)^*\right| d(G)^2\right)$.
4. $L(w_G, w_G) \in \Omega\left(\left|V(G)^*\right| + \left(d(G)^2\right)\right)$.

**Proof.** The proof follows easily from the previous work ■

## 5.1 Exercises

1. Define a bounded class of sandpile graphs, say $\mathcal{C}$, such that $d(G) \notin O\left(\sqrt{|G|}\right)$. How long are, on average, the critical avalanches occurring on $\mathcal{C}$-graphs?
2. Define a bounded class of sandpile graphs such that $d(G) \in O(\log(|G|))$. How short are the critical avalanches occurring on $\mathcal{C}$-graphs?
3. Are tight the upper and lower bounds established in this chapter? For which bounded classes are they tight?

# 6
# Dimension 1

In this chapter we study the algorithmic complexity of the one-dimensional sandpile model, that is: we study the complexity of The Abelian Sandpile Model, when restricted to the bounded class $\mathcal{L}_1$.

## 6.1 $GC\left[\mathcal{L}_1\right]$ belongs to $logDCFL$

In this section we prove that $GC\left[\mathcal{L}_1\right]$ belongs to $logDCFL$, which is the closure under logspace reduction of the class constituted by the deterministic context free languages. It is known that

$$NC^1 \subseteq logDCFL \subseteq AC^1 \subseteq NC^2$$

The class $logDCFL$ has an interesting machine characterization.

**Definition 55** *A pdTM is a logspace bounded Turing machine which has access to a pushdown stack.*

**Teorema 56** *(Sudborough's Theorem) $L \in logDCFL$ if and only if there exists a $pdTM$, say $\mathcal{M}$, such that $\mathcal{M}$ solves $L$.*

A proof of this theorem can be found in [S].

We will prove that the problem $GC\left[\mathcal{L}_1\right]$ can be solved using a $pdTM$, that is: we prove that $GC\left[\mathcal{L}_1\right]$ belongs to $logDCFL$. First at all we have to establish some basic facts concerning the dynamics of one-dimensional sandpiles.

**Lemma 57** *Let $f$ be a configuration on the one-dimensional lattice $\mathcal{L}_1^n$, suppose that there exists $i \in [n]$ such that:*

1. *$f(i) = 2$.*
2. *For any $j \neq i$ we have that $f(j) \leq 1$.*

*Furthermore we suppose that there exists $j_1 \lneqq i \lneqq j_2$ such that $f(j_1) = f(j_2) = 0$, and given $k \in \{j_1+1, ..., j_2-1\} - \{j\}$ we have that $f(k) = 1$. Then, $st_{\mathcal{L}_1^1}(f)$ is the configuration defined by*

$$st_{\mathcal{L}_1^1}(f)(v) = \begin{cases} 0 \text{ if } v = j_1 + j_2 - j \\ 1 \text{ otherwise} \end{cases}$$

Last lemma allows us to efficiently compute the relaxations of some very specific configurations. We ca use this as the basis of our predicting algorithm. Let $f, g$ be two critical configurations on $\mathcal{L}_1^n$, given $h = f + g$ we have that $Range(h) \subseteq \{0, 1, 2\}$, let $i_1 \lneqq i_2 \lneqq ... \lneqq i_k$ be the positions where $h$ takes the value 2, and let $t = h - e_{i_2} - ... - e_{i_k}$. Define a sequence $t_1, ..., t_k$ in the following way

$$\begin{aligned} t_1 &= st_{\mathcal{L}_1^n}(t) \\ t_2 &= st_{\mathcal{L}_1^n}(t_1 + e_{i_2}) \\ &\vdots \\ t_k &= st_{\mathcal{L}_1^n}(t_{k-1} + e_{i_k}) \end{aligned}$$

the abelianicity of the model implies that $st_{\mathcal{L}_1^n}(h) = t_k$.

Given $f, g$ two critical configurations on $\mathcal{L}_1^n$, we have that $Range(f+g) \subseteq \{0, 1\}$, also a possible description of $st_{\mathcal{L}_1^n}(f+g)$ is the ordered list of its null positions.

**Teorema 58** *$GC[\mathcal{L}_1]$ can be computed using a pdTM.*

**Proof.** Let $(n, f, g)$ be an instance of $GC[\mathcal{L}_1]$. We suppose that the underlying graph of the two-dimensional lattice is the linear graph $[n+1] \cup \{0\}$, also we add two new nodes to $\mathcal{L}_1^n$, the nodes 0 and $n+1$ which will be play the role of the sink. We suppose that given $f$ a configuration $f(0) = f(1) = 0$.

At the beginning of the computation the input is written in the input tape, we suppose that the input is a word $0x_1...x_n0$, where given $i \leq n$ the equality $(f+g)(i) = x_i$ holds, furthermore we suppose that the pushdown stack and the work tape are empty.

We observe that we can partition the set $\{1, ..., n\}$ into three sets as follows:

1. $T_0 = \{i : (f+g)(i) = 2\}$.

2. $N_0 = \{i : (f+g)(i) = 0 \; \& \; \forall j \left((f+g)(j) = 2 \Rightarrow i \lneqq j\right)\}$.
3. $M_0 = \{i : (f+g)(i) = 0 \; \& \; \exists j \left((f+g)(j) = 2 \; \& \; j \lneqq i\right)\}$.

Let $T_0 = \{i_1 \lneqq i_2 \lneqq ... \lneqq i_k\}$. We define $t = h - e_{i_2} - ... - e_{i_k}$, and we define a sequence $t_1, ..., t_k$ as before, that is:

$$\begin{aligned} t_1 &= st_{\mathcal{L}_1^n}(t) \\ t_2 &= st_{\mathcal{L}_1^n}(t_1 + e_{i_2}) \\ &\vdots \\ t_k &= st_{\mathcal{L}_1^n}(t_{k-1} + e_{i_k}) \end{aligned}$$

We know that $t_k = st_{\mathcal{L}_1^n}(f+g)$. Given $l \leq k$ we define

1. $T_l = \{i : t_l(i) = 2\}$.
2. $N_l = \{i : t_l(i)(i) = 0 \; \& \; \forall j \left(t_l(i)(j) = 2 \Rightarrow i \lneqq j\right)\}$.
3. $M_l = \{i : t_l(i)(i) = 0 \; \& \; \exists j \left(t_l(i)(j) = 2 \; \& \; j \lneqq i\right)\}$.

We observe that $T_k = M_k = \varnothing$. Also, in order to compute $t_k$ it is sufficient to compute $N_k$ ($N_k$ fully describes the configuration $t_k$). We try to compute $N_k$, to this end we compute the whole sequence

$$(N_1, T_1, M_1), ..., (N_k, T_k, M_k)$$

Finally we observe that $T_i$ and $M_i$ are determined by their first elements, (which we denote with the symbols $d_i$ and $m_i$), since, along the whole computation, we have access to the input (which is saved on the input tape).

The computation is divided in two stages, the first one, which we call initialization, consists in the computation of $h_1$.

*First Stage: Initialization*

1. Given $a_1, ..., a_r$ the elements of $N_0$ we write the word $a_1 \# a_2 \# ... \# a_r$ on the pushdown stack.
2. We compute $d_0$ and $m_0$, and we save these two numbers on the work tape, (using $O(\log(n))$ cells).

The second stage consists in the computation of $(N_i, d_i, m_i)$ from the previously computed triple $(N_{i-1}, d_{i-1}, m_{i-1})$.

*Second Stage: Computing the sequence*

Suppose we have computed the triple $(N_{i-1}, d_{i-1}, m_{i-1})$.

1. We compute $\alpha$, the maximum of $N_{i-1}$, which is the number written at the top of the pushdown stack.

2. We compute $z = m_{i-1} + \alpha - d_{i-1}$. (Note that $\alpha \lneqq z \lneqq m_{i-1}$)

3. If we suppose that there exists $t \in T_0$ such that $d_{i-1} \lneqq t \lneqq z$. Then, we have that $N_i = N_{i-1} - \{\alpha\}$ (we erase the number at the top of the pushdown stack), $m_i = z$ and $t_i = \min_{t \in T_0} \{d_{i-1} \lneqq t\}$. Otherwise, i.e. if there not exists $t \in T_0$ such that $d_{i-1} \lneqq t \lneqq z$, we have that $N_i = (N_{i-1} \cup \{z\}) - \{\alpha\}$ (we erase the number at the top of the stack and after that we write $z$), $t_i = \min_{t \in T_0} \{d_{i-1} \lneqq t\}$ and $m_i = \min_{j \in M_0} \{j \gneqq m_{i-1}\}$.

4. We stop when $T_i$ becomes an empty set.

It is easy to check that if $b_1\#...\#b_r$ is the word written on the pushdown stack at the end of the computation, then $N_k = \{b_1, ..., b_r\}$. Thus, we have computed the stabilization of $f + g$ using a $pdTM$. ■

**Corollary 59** *The problems $SPP[\mathcal{L}_1]$, $CSV[\mathcal{L}_1]$, $MC[\mathcal{L}_1]$, $MC^*[\mathcal{L}_1]$, $EC[\mathcal{L}_1]$, $GC[\mathcal{L}_1]$, $I[\mathcal{L}_1]$, $SPA[\mathcal{L}_1]$ and $RR[\mathcal{L}_1]$ belong to $NC^4$.*

**Remark 60** *It can be proved that $SPP[\mathcal{L}_1]$ also belongs to logDCFL.*

## 6.2 $SPA[\mathcal{L}_1]$ is $TC^0$-hard

Let $\mathcal{L}_1^{3n}$ be the one-dimensional undirected sandpile lattice on $\{1, ..., 3n\}$, (we can add the nodes 0, $3n+1$ and assign to them the role of sink nodes). Suppose that $g$ is a configuration on $\mathcal{L}_1^{3n}$ which satisfies the following three conditions.

1. If $i \leq n$, then we have $g(i) = 0$.

2. If $i \in \{n+1, ..., 2n\}$, then we have $g(i) \in \{1, 2\}$.

3. If $i \geq 2n+1$, then we have $g(i) = 0$.

Let $\|g\| = \sum_i g(i) \leq 2n$

**Teorema 61** *There exist numbers $i, j \in \{0, 1, ..., 3n+1\}$ such that*

1. *$i \lneqq j$ and $j - i \in \{\|g\|, \|g\| - 1\}$*

2. *If $k \notin \{i, i+1, ..., j\}$, then $st_{\mathcal{L}_1^{3n}}(g)(k) = 0$.*

3. *If $k \in \{i, i+1, ..., j\}$, then $st_{\mathcal{L}_1^{3n}}(g)(k) \in \{0, 1\}$; and there exists at most one $k$ such that $i \lneqq k \lneqq j$ and $st_{\mathcal{L}_1^{3n}}(g)(k) = 0$.*

**Proof.** Let $i_1 \leq i_2 \leq ... \leq i_k$ be the positions where the value of $g$ is equal to 2. Let $g_0$ be the configuration which takes the value 1 on the set $\{n+1, ..., 2n\}$ and the value 0 on its complement. Note that

$$g = g_0 + e_{i_1} + ... + e_{i_k}$$

The abelian property of the *abelian* sandpile model (22) implies that

$$st_{\mathcal{L}_1^{3n}}\left(g\right) = st_G\left(g_{k-1} + e_{i_k}\right)$$

where $g_1 = st_{\mathcal{L}_1^{3n}}\left(g_0 + e_{i_1}\right)$ and given $g_{i_r}$ we have that

$$g_{i_{r+1}} = st_{\mathcal{L}_1^{3n}}\left(g_{i_r} + e_{i_{r+1}}\right)$$

First at all we try to compute $g_1$. It is easy to check that $g_1$ is a configuration constituted by a zero floating in a connected sea of ones, and that the position of the isolated zero is the mass center of the configuration $g_0 + e_{i_1}$, that is: we have a position $j_1 \in \{n+1, ..., 2n\}$ such that $g_1\left(j\right) = 0$. Furthermore, we have that if $j \in \{n+1, ..., 2n\} - \{j_1\}$, then $g_1\left(j\right) = 1$; and either $g_1\left(n\right) = 1$ or $g_1\left(2n+1\right) = 1$.

Now, we try to compute $g_2$. If $j_1 = i_2$, then there exists an interval $I_1 \supseteq \{n+1, ..., 2n\}$ such that $g_1 + e_{i_2}$ takes the value 1 on $I_1$ and the value 0 out of $I_1$. In this case $g_1 + e_{i_2}$ is already a stable configuration and is equal to $g_2$. If $j_1 \neq i_2$, then there exists an interval $I_1 \supseteq \{n+1, ..., 2n\}$ such that $g_1 + e_{i_2}$ takes the value 0 out of $I_1$; $\left(g_1 + e_{i_2}\right)\left(j_1\right) = 0$; $\left(g_1 + e_{i_2}\right)\left(i_2\right) = 2$; and $g_1 + e_{i_2}$ takes the value 1 on any other point of $I_1$. So, the configuration $g_1 + e_{i_2}$ looks like a zero and a two floating in a connected and isolated sea of ones. It is easy to check that $g_2 = st_{\mathcal{L}_1^{3n}}\left(g_1 + e_{i_2}\right)$ is a stable configuration of one of the following two types:

1. (type 1) There exists an interval $I_2 \supseteq \{n+1, ..., 2n\}$ such that the configuration $st_{\mathcal{L}_1^{3n}}\left(g_1 + e_{i_2}\right)$ takes the value 0 out of $I_2$ and the value 1 on $I_2$. Furthermore, the length of $I_2$ is $n+2$.

2. (type 2) There exists an interval $I_2 \supseteq \{n+1, ..., 2n\}$ such that the configuration $st_{\mathcal{L}_1^{3n}}\left(g_1 + e_{i_2}\right)$ takes the value 0 out of $I_2$ and the value 1 on $I_2 - \{x\}$, where $x \in I_2$ and $st_{\mathcal{L}_1^{3n}}\left(g_1 + e_{i_2}\right)\left(x\right) = 0$. Furthermore, the length of $I_2$ is $n+3$.

At this point, it should be clear that we can use an inductive argument to prove that for any $j \leq k$, the configuration $g_j$ is a configuration of one of the following two types:

1. (type 1) There exists an interval $I_j \supseteq \{n+1, ..., 2n\}$ such that $g_j$ takes the value 0 out of $I_j$ and the value 1 on $I_j$. Furthermore, the length of $I_j$ is $n+j$.

2. (type 2) There exists an interval $I_j \supseteq \{n+1, ..., 2n\}$ such that $g_j$ takes the value 0 out of $I_j$ and the value 1 on $I_j - \{x\}$, where $x \in I_j$ and $g_j(x) = 0$. Furthermore, the length of $I_j$ is $n + j + 1$.

If we take $j = k$ we obtain the theorem. ■

We are ready to prove the main theorem of this subsection. We will prove that the computation of the majority function is constant depth reducible to $SPA[\mathcal{L}_1]$.

Given $(x_1, ..., x_n)$, we have that $Maj(x_1, ..., x_n) = 1$ if and only if $\sum x_i \geq \lfloor \frac{n}{2} \rfloor + 1$. Note that $Maj(x_1, ..., x_n, x_{n+1} = x_1, ..., x_{2n} = x_n) = 1$ if and only if $\sum x_i \geq n + 1$ if and only if $\sum x_i \gneqq n$.

**Teorema 62** *$SPA[\mathcal{L}_1]$ is $TC^0$-hard.*

**Proof.** We show that the computation of the majority function is constant depth reducible to $SPA[\mathcal{L}_1]$. Suppose that $x = (x_1, ..., x_n)$ is a boolean vector. Let $m = 2n$ and let $(y_1, ..., y_m) = (x_1, ..., x_n, x_1, ..., x_n)$. We define a configuration $g_x$ on $\{0, 1, ..., 3m+1\}$ as follows

$$g_x(i) = \begin{cases} y_j + 1 \text{ if } i = m + j \text{ and } j \in \{1, ..., m\} \\ 0 \text{ else} \end{cases}$$

Note that $g_x$ satisfies the conditions in the statement of theorem 61. Let us call the *shadow* of $g_x$ to the area out of $\{m+1, ..., 2m\}$ that will be filled with chips after the relaxation process. If $Maj(x) = 1$, then the shadow of $g_x$ will be large, it will fill at least $n + 2$ positions. On the other side, if $Maj(x) = 0$, then the shadow of $g_x$ will be small, it will fill at most $n + 1$ positions. Let $A_m$ be equal to the set

$$\{(i,j) : 0 \leq i \leq m \ \& \ j \gneqq 2m \ \& \ j - i \geq m + n + 1\}$$

Note that $Maj(x_1, ..., x_n) = 1$ if and only if

$$\bigvee_{(i,j) \in A_m} (((G_m, g, i+1) \in SPA[\mathcal{L}_1]) \wedge ((G_m, g, j-1) \in SPA[\mathcal{L}_1]))$$

Thus, we have proven that we can compute the majority function using a $D\log time$ uniform family of depth-three circuits, with an or gate on the top; a second layer composed by and-gates and a first layer composed by $SPA[\mathcal{L}_1]$ oracle gates. Therefore, we have that $SPA[\mathcal{L}_1]$ is $TC^0$-hard. ■

**Corollary 63** *$SPP[\mathcal{L}_1]$ is $TC^0$-hard, and $SPP[\mathcal{L}_1]$ belongs to $AC^1$ but given $\epsilon \gneqq 0$ we have that $SPP[\mathcal{L}_1] \notin AC^{1-\epsilon}$.*

## 6.3 A long remark: one-dimensional critical avalanches

In this section we will establish some facts concerning the dynamics of one-dimensional sandpiles and one-dimensional critical avalanches.

**Teorema 64** *Let $n$ be a natural number, we have*

1. $L\left(w_n^1, w_n^1\right) \in \Omega\left(n^2\right)$.
2. *There exists a positive constant $K$ such that*

$$\Pr_{f,g\in\mathcal{K}\left(\mathcal{L}_n^1\right)}\left[L\left(f,g\right) \geq Kn^2\right] \geq \frac{1}{8}$$

**Proof.** It follows from theorem 49 that $L\left(w_n^1, w_n^1\right) \in \Omega\left(\left|\mathcal{L}_n^1\right| + d\left(\mathcal{L}_n^1\right)^2\right)$, we observe that $\left|\mathcal{L}_n^1\right| = n$ and $\left(d\left(\mathcal{L}_n^1\right)\right)^2 = \left(\left\lceil \frac{n}{2}\right\rceil\right)^2 \approx \frac{n^2}{4}$. Items 2 is consequences of theorem 49. ■

Also, we have that most critical avalanches are very long, they have a length which is at least cuadratic with respect to the size of the lattice, but in despite of this we can predict the final state of those avalanches in time $O\left(\log^2\left(n\right)\right)$, that is: ¡predicting is possible! It is possible to predict in short time the evolution of long sandpile dynamics.

## 6.4 Exercises

1. Prove Sudborough's theorem.
2. Prove lemma 57.
3. Look for the definition of $AC^{1-\epsilon}$. Prove that $GC\left[\mathcal{L}_1\right]$ doesn't belong to $AC^{1-\epsilon}$ for any $\epsilon \gneqq 0$.

# 7 Dimension 2

In this chapter we study the two-dimensional sandpile model, that is: we study the restriction of The Abelian Sandpile Model to two-dimensional square lattices.

## 7.1 The hardness of two-dimensional sandpile prediction problems

In this section we prove that $SPA[\mathcal{L}_2]$ is $NC^1$ hard, to this end we prove that the problem $MCVP[\mathcal{P}]$ is $\log space$ reducible to $SPA[\mathcal{L}_2]$. Remember that $MCVP[\mathcal{P}]$ (*The Planar Monotone Circuit Value Problem*) is the problem defined by:

**Problem 65** *(Planar Monotone Circuit Value Problem)*

- *Input:* $(\mathcal{C}, \mu)$, *where* $\mathcal{C}$ *is a planar monotone boolean circuit and* $\mu$ *is a valuation.*
- *Problem: Compute* $\mathcal{C}(\mu)$.

It is important to remark that $MCVP[\mathcal{P}]$ is $NC^1$ hard under *logspace* reductions, also if we define a *logspace* reduction of $MCVP[\mathcal{P}]$ in $SPA[\mathcal{L}_2]$, we prove the $NC^1$ hardness of $SPA[\mathcal{L}_2]$. Therefore, our goal is to define a *logspace* algorithm which, on input $(\mathcal{C}, \mu)$ an instance of $MCVP[\mathcal{P}]$, computes an instance of $SPA[\mathcal{L}_2]$, say $(\mathcal{L}_2^n, f, v)$, such that $\mathcal{C}(\mu) = 1$ if and only if $SC_f(v) \geq 1$. It is important to remark that we can suppose w.l.o.g. that $\mathcal{C}$ is a synchronous planar circuit whose fan out and fan in are bounded by two.

**Teorema 66** *$SPA[\mathcal{L}_2]$ is $NC^1$ hard.*

**Proof.** Let $(\mathcal{C}, \mu)$ be an instance of $MCVP[\mathcal{P}]$ such that $\mathcal{C}$ is a synchronous planar circuit whose fan in and fan out are bounded by two. Let $G_{\mathcal{C}}$ be the underlying graph of $\mathcal{C}$, we can compute, using logarithmic space, an embedding of $G_{\mathcal{C}}$ into $\mathcal{L}_2^{p(n)}$, where $n$ is the size of $G_{\mathcal{C}}$ and $p(X)$ is a suitable polynomial. Furthermore, we can suppose that the image of $G_{\mathcal{C}}$ is fully contained in the interior of $\mathcal{L}_2^{p(n)}$. We say that a node $v \in V\left(\mathcal{L}_2^{p(n)}\right)$ belongs to the image of such an embedding if and only if either $v$ is he image of some node $w \in V(G_{\mathcal{C}})$, or $v$ is located on the image of one of the edges of $G_{\mathcal{C}}$ (the embedding sends nodes of $G_{\mathcal{C}}$ in nodes of $\mathcal{L}_2^{p(n)}$ and edges of $G_{\mathcal{C}}$ in simple walks of $\mathcal{L}_2^{p(n)}$). In the former case we say that $v$ is a gate-node and in the later we say that $v$ is a wire node. We note that given $v$ a gate-node, $v$ has exactly one preimage which will be denoted with the symbol $w_v$. Given $o$ the output node of $G_{\mathcal{C}}$, we use the symbol $v_o$ to denote its image. Now we define a configuration on $\mathcal{L}_2^{p(n)}$ :

1. If $v$ is a gate-node and $w_v$ is an and-gate we set $f(v) = 2 = \deg(v) - 2$.
2. If $v$ is a gate-node and $w_v$ is an or-gate we set $f(v) = 3 = \deg(v) - 1$.
3. If $v$ is a wire-node we set $f(v) = 3 = \deg(v) - 1$.
4. If $v$ is a gate node, $w_v$ is an input gate and $\mu(w_v) = 1$, we set $f(v) = 4$.
5. If $v$ is a gate node, $w_v$ is an input gate and $\mu(w_v) = 0$, we set $f(v) = 0$.

The main idea of such a construction is that we are identifying the following two facts:

1. Gate $w_v$ evaluates to 1 (0); node $v$ fires (doesn't fires).
2. A true signal flows through a given wire; chips flows through the corresponding simple walk in the lattice $\mathcal{L}_2^{p(n)}$.

The construction will work if we add a last ingredient called diode. Diodes are gadgets designed to ensure that the flow of chips in the lattice is a directed flow analogous to the flow of true signals in the circuit, that is: diodes are constructions designed to ensure that chips flow in the right direction and that spurious signals are not produced.

A diode has the following structure

$$\begin{array}{cccccccc} & & & 3 & 3 & & & \\ 3 & 3 & 3 & 3 & 2 & 3 & 3 & 3 \end{array}$$

*Chips can flow from left to right*

*but not vice versa*

We built a diode on any lattice-wire (simple path representing a circuit wire), if we don't have enough space (i.e. if there is either overlapping of diodes or overlapping of diodes and wires) we can subdivide the lattice as many times as necessary eliminating the overlapping of our constructions. Finally after adding all the diodes, if $v$ doesn't belongs to the subgraph of $\mathcal{L}_2^{p(n)}$ simulating the circuit (that is: $v$ is neither a gate node nor a wire node) we set $f(v) = 0$.

It is not difficult to check that

$$SC_f(v_0) \geq 1 \text{ if and only if } \mathcal{C}(\mu) = 1$$

Then, we have that $MCVP[\mathcal{P}]$ is *logspace* reducible to $SPA[\mathcal{L}_2]$ since the whole construction can be computed in *logspace* ■

## 7.2 Two-dimensional critical avalanches

In this short section we will say some things related to the statistics of two-dimensional critical avalanches, it will allows us to analyze the average-perfomance of simulation algorithms, when they are employed to solve the problem $GC[\mathcal{L}_2]$.

**Teorema 67** *There exist positive constants $C, D$ such that*

1. *For all $f, g \in \mathcal{K}(\mathcal{L}_2^n)$ we have that $L(f,g) \geq \frac{n^2-16n}{4}$.*
2. *$L(w_n^2, w_n^2) \leq Cn^4 \in O\left(|\mathcal{L}_2^n|^2\right)$.*
3. *$L(w_n^2, w_n^2) \geq Dn^3 \in \Omega\left(|\mathcal{L}_2^n|^{1.5}\right)$.*

**Proof.** Item 1 follows directly from theorem 45. Item 2 is a consequence of theorem 47 and the following fact: given $n \geq 1$ we have that $d\left(\mathcal{L}_n^2\right) = \lfloor \frac{n}{2} \rfloor$. We prove Item 3. It follows from the proof of theorem 49 that $L\left(w_{\mathcal{L}_2^n}, w_{\mathcal{L}_2^n}\right) \geq \sum_{i=1}^{d(\mathcal{L}_2^n)} |M_i(\mathcal{L}_2^n)|$. Note that for any $i \leq d(\mathcal{L}_2^n)$ the size of $M_i(\mathcal{L}_2^n)$ is equal to $i^2$. Also, we have

$$L\left(w_{\mathcal{L}_2^n}, w_{\mathcal{L}_2^n}\right) \geq \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} i^2 \in \Omega\left(n^3\right) = \Omega\left(|\mathcal{L}_2^n|^{1.5}\right)$$

■

Let $\mathcal{SA}$ be the naive (sequential) sandpile automata simulation algorithm, and let $\mathcal{B}$ be the parallel sandpile automata simulation algorithm (we topple all the unstable nodes at once). We will use the symbol $t_{\mathcal{SA}}^{(2)}(n, f, g)$ to denote the running time of $\mathcal{SA}$ on input $\left(\mathcal{L}_n^2, f, g\right)$, we define $t_{\mathcal{B}}^{(2)}(n, f, g)$ accordingly.

**Teorema 68** *Let $\mathcal{L}$ be the class of square sandpile grids, we have that:*

1. *Given $f, g \in \mathcal{K}(\mathcal{L}_n)$ we have that $t_{\mathcal{SA}}^{(2)}(n, f, g) \geq \frac{n^2-64n}{4} \in \Omega\left(\left|\mathcal{L}_n^2\right|\right)$.*
2. *There exists a constant $C$ such that for any $n \geq 1$*

$$\Pr_{f,g\in\mathcal{K}(\mathcal{L}_n)}\left[t_{\mathcal{SA}}^{(2)}(n, f, g) \geq C\left|\mathcal{L}_n^2\right|^{1.5}\right] \geq \frac{1}{32}$$

3. *There exists a constant $C$ such that for any $n \geq 1$*

$$\Pr_{f,g\in\mathcal{K}(\mathcal{L}_n)}\left[t_{\mathcal{B}}^{(2)}(n, f, g) \geq C\left|\mathcal{L}_n^2\right|^{0.5}\right] \geq \frac{1}{32}$$

**Proof.** We have already proven items 1 and 2. We prove item 3. Let $f, g$ be two elements of $\mathcal{K}\left(\mathcal{L}_n^2\right)$ such that $L(f, g) \geq C\left|V\left(\mathcal{L}_n^2\right)^*\right|^{1.5}$. It implies that there exists a node $v$ which is toppled at least $C\left|V\left(\mathcal{L}_n^2\right)^*\right|^{0.5}$ times. If we are using the parallel updating protocol (that is, if we are running the algorithm $\mathcal{B}$) the topplings performed at $v$ have to be performed at different times, and it implies that $t_{\mathcal{B}}^{(2)}(n, f, g) \geq C\left|V\left(\mathcal{L}_n^2\right)^*\right|^{0.5}$. Thus, we have that

$$\Pr_{f,g\in\mathcal{K}(\mathcal{L}_n^2)}\left[t_{\mathcal{B}}^{(2)}(n, f, g) \geq C\left|\mathcal{L}_n^2\right|^{0.5}\right] \geq \frac{1}{2D^2}$$

■

## 7.3 Exercises

1. Define the class $\mathcal{T}$ of triangular sandpile lattices. Is $\mathcal{T}$ a bounded class?
2. Define the class $\mathcal{H}$ of honeycomb sandpile lattices. Is $\mathcal{H}$ a bounded class?
3. Are there classes of two-dimensional regular lattices other than $\mathcal{T}$, $\mathcal{H}$ and $\mathcal{L}_2$?

# 8
# Dimension 3

In this chapter we study the restriction of The Abelian Sandpile Model to three-dimensional cubic lattices.

## 8.1 $RR[\mathcal{L}_3]$ is $P$-complete

If we would adapt Moore's construction to the three-dimensional setting, we would obtain a reduction of the *Monotone Circuit Value Problem* ($MCVP$ for short) into $SPA[\mathcal{L}_3]$, that is: we can prove that $SPA[\mathcal{L}_3]$ is $P$-complete. In this section we prove something stronger, we prove that $RR[\mathcal{L}_3]$ (the easiest of our three-dimensional algorithmic problems) is $P$-complete.

**Teorema 69** *$RR[\mathcal{L}_3]$ is $P$-complete.*

**Proof.** Let $(\mathcal{C}, \mu)$ be an instance of $MCVP$, we suppose that $\mathcal{C}$ is synchronous, has fan in and fan out bounded by two, and we suppose that $\mathcal{C}$ is a $n$-size circuit with $m$ input gates. If we use the basic components of Moore's construction (fuses, diodes, gates and duplicators) we can built a *sandpile circuit* into a three-dimensional lattice $\mathcal{L}_3^{p(n)}$ (where $p(X)$ is a suitable polynomial) in such a way that the nodes of the circuit are contained in the interior of $\mathcal{L}_3^{p(n)}$. Let $\mathcal{C}^*$ be the subgraph of $\mathcal{L}_3^{p(n)}$ whose nodes are the nodes of the circuit. We can claim that any node of $\mathcal{C}^*$ has a neighbor out of $\mathcal{C}^*$, moreover we can claim that the complement of $\mathcal{C}^*$ is a connected subgraph of $\mathcal{L}_3^{p(n)}$. Given $v \in V(\mathcal{C}^*)$ we assign to $v$ two positive integers in the following way:

1. $k_v = \left|\left\{w \in V\left(\mathcal{C}^*\right) : \{v,w\} \in E\left(\mathcal{L}_3^{p(n)}\right)\right\}\right|$.
2. $r_v = \left|\left\{w \notin V\left(\mathcal{C}^*\right) : \{v,w\} \in E\left(\mathcal{L}_3^{p(n)}\right)\right\}\right|$.

Now, we define a configuration $f_{(\mathcal{C},\mu)}$ on $\mathcal{L}_3^{p(n)}$ :

1. If $v \notin V\left(\mathcal{C}^*\right)$ we set $f_{(\mathcal{C},\mu)}(v) = 5$.
2. If $v$ is a fuse-node we set $f_{(\mathcal{C},\mu)}(v) = 5 - r_v$.
3. If $v$ is an or-gate we set $f_{(\mathcal{C},\mu)}(v) = 5 - r_v$.
4. If $v$ is an and-gate we set $f_{(\mathcal{C},\mu)}(v) = 4 - r_v$.
5. If $v$ is an input gate corresponding to the variable $X$ and $\mu(X) = 1$, then we set $f_{(\mathcal{C},\mu)}(v) = 6 - r_v$.
6. If $v$ is an input gate corresponding to the variable $X$ and $\mu(X) = 0$, then we set $f_{(\mathcal{C},\mu)}(v) = 0$.

Now, we prove that $f_{(\mathcal{C},\mu)}$ is critical if and only if $\mathcal{C}(\mu) = 1$. First at all we remember that $f_{(\mathcal{C},\mu)}$ is critical if and only if for any $v \in V\left(\mathcal{L}_3^{p(n)}\right)^*$ we have $SC_{f+e_{\delta\left(\mathcal{L}_3^{p(n)}\right)}}(v) \geq 1$.

We begin by adding one chip to any node on the border of $\mathcal{L}_3^{p(n)}$, then we fire those nodes. We continue firing any unstable node out of $V\left(\mathcal{C}^*\right)$.

**Claim**. We can fire all the nodes out of $V\left(\mathcal{C}^*\right)$, before firing the first node in $V\left(\mathcal{C}^*\right)$.

(proof of the claim). Remember that for any node out of $V\left(\mathcal{C}^*\right)$, say $v$, we have that $f_{(\mathcal{C},\mu)}(v) = 5$. Also, $v$ becomes unstable if and only if at least one neighbor of $v$ is fired. Note that after adding the border configuration, we can fire all the nodes in the border of $\mathcal{L}_3^{p(n)}$, without firing nodes in $V\left(\mathcal{C}^*\right)$ (we are supposing that $V\left(\mathcal{C}^*\right)$ is contained in the interior of $\mathcal{L}_3^{p(n)}$). We prove the claim using an inductive argument with respect to the distance to the border.

- (distance zero) The nodes on the border can be fired.
- (distance $i$) We suppose that any node out of $V\left(\mathcal{C}^*\right)$, whose distance to the border is less than or equal to $i$, can be fired before firing the first node in $V\left(\mathcal{C}^*\right)$.
- (distance $i+1$) Let $v$ be a node such that the distance from $v$ to the border is equal to $i+1$. There exists $j$ such that $\{i,j\} \in E\left(\mathcal{L}_3^{p(n)}\right)$, $j \in \left(V\left(\mathcal{C}^*\right)\right)^c$ and the distance from $j$ to the border is equal to $i$. The inductive hypothesis says us that node $j$ can be fired before firing the

first node of $V\left(\mathcal{C}^*\right)$. We observe that after firing $j$, node $i$ becomes unstable. Also, we can fire $i$ just after firing $j$. Thus, we can fire $i$ before firing any node in $V\left(\mathcal{C}^*\right)$.

Thus, we can fire all the nodes out of $V\left(\mathcal{C}^*\right)$, before firing nodes in $V\left(\mathcal{C}^*\right)$. We observe that if we fire all the nodes out of $V\left(\mathcal{C}^*\right)$, the nodes in $V\left(\mathcal{C}^*\right)$ (which are the nodes that have been not fired) get the right values, that is: after firing the complement of $V\left(\mathcal{C}^*\right)$, the subgraph $\mathcal{C}^*$ carries a configuration such that all the nodes of $V\left(\mathcal{C}^*\right)$ are fired in the relaxation process if and only if $\mathcal{C}\left(\mu\right) = 1$. Therefore, we have that $\mathcal{C}\left(\mu\right) = 1$ if and only if for any $v$, $SC_{f+e_{\delta\left(\mathcal{L}_3^{p(n)}\right)}}\left(v\right) \geq 1$ if and only if $f$ is critical. Thus, we have proven that $MCVP$ is logspace reducible to $SPP\left[\mathcal{L}_3\right]$ ■

## 8.2 Strict $P$-completeness of $SPP\left[\mathcal{L}_3\right]$

If $NC \neq P$ and $L$ is a $P$-complete problem there not exist polylogarithmic time parallel algorithms computing the problem $L$. Also, the notion of $P$-completeness allows us to discard the existence of polylogarithmic time algorithms solving a given problem. If we want to discard the existence of sublinear time algorithms computing a given problem, we have to use a stronger notion. Anne Condon [C] introduced the notion of strict $P$-completeness, which can be used to decide questions concerning the existence-nonexistence of sublinear time algorithms.

**Definition 70** *Given $L, \Omega$ Two languages in $P$ a honest $NC$ reduction of $L$ in $\Omega$ is a $NC$ reduction $\mathcal{N}$ for which there exists $k \geq 1$ such that for any $x$ input of $L$, large enough, and for any query $y$, computed by $\mathcal{N}$ on input $x$, the inequality $\left|y\right| \geq \left|x\right|^k$ holds.*

**Definition 71** *Given $f : \mathbb{N} \rightarrow \mathbb{N}$ a nondecreasing function and given $L \in P$, we say that $L$ is $f$-hard for $P$ if and only if for all $\Omega \in P$, for any sequential algorithm $\mathcal{M}$ computing $\Omega$ and for all $\varepsilon \gneqq 0$ there exists a honest $NC$-reduction $\mathcal{N}$ from $\Omega$ into $L$, such that for any instance $x$ of $\Omega$ and for any query $y$ computed by $\mathcal{N}$, on input $x$, we have that*

$$f\left(\left|y\right|\right) = O\left(t_{\mathcal{M}}\left(\left|x\right|\right)\left|x\right|^{\varepsilon}\right)$$

*where $t_{\mathcal{M}}$ is the running time of $\mathcal{M}$. We say that $L$ is $f$-complete for $P$ if and only if $L$ is $f$-hard for $P$ and $L$ can be solved in time $O\left(f\right)$.*

From a naive point of view we have that a problem $L$ is $f$-hard for $P$ if and only if any algorithm computing $L$ has a running time $\Omega\left(f\right)$. The three theorems listed below are the core of the theory of strict $P$-completeness.

**Teorema 72** *If $L$ is $f$-hard for $P$ and there exists $\varepsilon \gneqq 0$ such that $L$ can be solved in time $f(|x|)|x|^{-\varepsilon}$. Then, any problem $\Omega \in P$ can be solved in time $\widetilde{O}(t_{\Omega}(n) n^{-\varepsilon})$, where $t_{\Omega}$ is the sequential time of $\Omega$.*

**Proof.** Let $\Omega$ be a problem in $P$, and let $\mathcal{M}$ be an algorithm solving $L$ in time $t_{\mathcal{M}}(n)$ and let $\mathcal{N}$ be a honest $NC$ reduction of $\Omega$ in $L$ such that for any query $y$, computed by $\mathcal{N}$ on input $x$, we have that

$$f(|y|) = O\left(t_{\mathcal{M}}(|x|)|x|^{\varepsilon}\right) \text{ and } |y| \geq |x|^{\frac{1}{k}}$$

where $\varepsilon \gneqq 0$ and $k \geq 1$. Let $\mathcal{B}$ be a parallel algorithm solving $L$ in time $f(n)|n|^{-2k\varepsilon}$. Let $\mathcal{A}$ be the parallel algorithm defined by:
On input $x$ algorithm $\mathcal{A}$ works as follows:

- Algorithm $\mathcal{A}$ simulates algorithm $\mathcal{N}$ on input $x$. Furthermore, given $y$ a query computed by $\mathcal{N}$ on input $x$, algorithm $\mathcal{A}$ simulates algorithm $\mathcal{B}$ on input $y$ instead of asking the oracle for $L$.

It is clear that algorithm $\mathcal{A}$ solves problem $\Omega$. Now, we have to upperbound the running time of $\mathcal{A}$. First at all we note that $t_{\mathcal{A}}(x)$, the running time of $\mathcal{A}$ on input $x$, is upperbounded by $t_{\mathcal{N}}(x) + \max_{y \in Q_{\mathcal{N}}(x)} \{t_{\mathcal{B}}(y)\}$, where $t_{\mathcal{N}}(x)$ denotes the running time of $\mathcal{N}$, $t_{\mathcal{B}}(y)$ denotes the running time of $\mathcal{B}$ on input $y$ and $Q_{\mathcal{N}}(x)$ denotes the set of queries computed by $\mathcal{N}$ on input $x$. We note that $t_{\mathcal{N}}$ is polylogarithmic, also we focus our attention on the term $\max_{y \in Q_{\mathcal{N}}(x)} \{t_{\mathcal{B}}(y)\}$. First at all we remember that: $|y| \geq |x|^{\frac{1}{k}}$ and for all $y \in Q_{\mathcal{N}}(x)$ the relation $f(|y|) \in O\left(t_{\mathcal{M}}(|x|)|x|^{\varepsilon}\right)$ holds. Then, we have that there exist two constants $C, D$ such that

$$\begin{aligned}
\max_{y \in Q_{\mathcal{N}}(x)} \{t_{\mathcal{B}}(y) t_{\mathcal{N}}(|x|)\} &\leq \max_{y \in Q_{\mathcal{N}}(x)} \left\{ f(|y|) \left(\left|x^{\frac{1}{k}}\right|\right)^{-2k\varepsilon} t_{\mathcal{N}}(|x|) \right\} \\
&\leq C\left(t_{\mathcal{M}}(|x|)|x|^{\varepsilon}|x|^{-2\varepsilon} t_{\mathcal{N}}(|x|)\right) + D \\
&\leq C\left(t_{\mathcal{M}}(|x|)|x|^{-\varepsilon} t_{\mathcal{N}}(|x|)\right) + D
\end{aligned}$$

Thus, we have that $t_{\mathcal{A}}(n) \in \widetilde{O}(t_{\mathcal{M}}(n) n^{-\varepsilon})$. Then, we have that $\Omega$ can be solved in time $\widetilde{O}(t_{\Omega}(n) n^{-\varepsilon})$. ■

Given $L$ and $\Omega$ two problems in $P$, and given $\mathcal{M}$ a reduction from $L$ into $\Omega$, we define a function $s_{\mathcal{M}}$ as follows

$$s_{\mathcal{M}}(n) = \max\{|\mathcal{M}(y)| : \exists x (|x| \leq n \;\&\; y \in Q_{\mathcal{M}}(x))\}$$

and we define a second function $s_{\mathcal{M}}^{-1}$ in the following way

$$s_{\mathcal{M}}^{-1}(n) = \max\{m : s_{\mathcal{M}}(m) \leq n\}$$

**Teorema 73** *If $L$ is $f$-hard for $P$ and there exists a $NC$-reduction $\mathcal{M}$ of $L$ in $\Omega$, then $\Omega$ is $f \circ s_{\mathcal{M}}^{-1}$-hard for $P$.*

**Proof.** Let $\Psi$ be a language in $P$ whose sequential running time is equal to $t(n) \in \Omega(n)$ and let $\varepsilon \gneqq 0$. There exists a honest $NC$ reduction of $\Psi$ in $L$, say $\mathcal{N}$, such that for any $x$ instance of $\Psi$ and for any $y \in Q_{\mathcal{N}}(x)$ we have that $f(|y|) \in O\left(t(|x|)|x|^{\varepsilon}\right)$. Let $\mathcal{M} \circ \mathcal{N}$ be the composition of $\mathcal{M}$ and $\mathcal{N}$. It is easy to check that $\mathcal{M} \circ \mathcal{N}$ is a honest $NC$ reduction. Given $x$ an instance of $\Psi$ and given $z$ a query computed by $\mathcal{M} \circ \mathcal{N}$ on input $x$ we have that:

1. There exists $y \in Q_{\mathcal{N}}(x)$ such that $z$ is a query computed by $\mathcal{M}$ on input $y$.
2. Since $L$ is $f$-hard we have that for any $y \in Q_{\mathcal{N}}(x)$, the containment $f(|y|) \in O\left(t(|x|)|x|^{\varepsilon}\right)$ holds.
3. $|z| \leq s_{\mathcal{M}}(|y|)$.

Thus, we have

$$f\left(s_{\mathcal{M}}^{-1}(|z|)\right) \leq f\left(s_{\mathcal{M}}^{-1}\left(s_{\mathcal{M}}(|y|)\right)\right) = f(|y|) \in O\left(t(|x|)|x|^{\varepsilon}\right)$$

■

In order to use the theory we need at least one problem, for which some type of strictly $P$-hardness have been already established. Consider the following problem

**Problem 74** *(SCVP; Square circuit value problem)*

- *Input: $(C, n, f)$, where $C$ is a boolean circuit of size $n$, $f$ is a valuation for $C$ and $n$ is a square. Furthermore, we demand that $C$ is a synchronic, monotone boolean circuit of depth $\sqrt{n}$ such that each one of the $\sqrt{n}$ levels of $C$ is constituted by $\sqrt{n}$ gates, and all the inner gates have fan in and fan out equal to* 2.
- *Problem: Decide if $C$ accepts $f$.*

**Teorema 75** *The problem $SCVP$ is $\sqrt{n}$-hard for $P$.*

The proof can be found in [C]. The theorem says that the problem $SCVP$ plays a role in the theory of strict $P$-completeness which is analogous to the role played by $CVP$ (The Circuit Value Problem) in the theory of $P$-completeness.

**Teorema 76** *$SPP[\mathcal{L}_3]$ is $\sqrt[6]{n}$-hard for $P$.*

**Proof.** We use the symbol $\mathcal{N}$ to denote Moore's reduction of $MCVP$ in $SPP[\mathcal{L}_3]$. Given $(\mathcal{C}, \mu)$ an instance of $MCVP$, we have that $\mathcal{N}(\mathcal{C}, \mu)$ is equal to a pair $(G_{\mathcal{C}}, f_{\mathcal{C},\mu})$ such that $G_{\mathcal{C}}$ is a cubic lattice and $f_{\mathcal{C},\mu}$ is a configuration on $G_{\mathcal{C}}$, (which can be expressed as a sum of two stable configurations. Also, $\mathcal{N}$ is a reduction of $MCVP$ in $MC[\mathcal{L}_3]$). In the definition of the reduction $\mathcal{N}$, we suppose that any input of $\mathcal{N}$ is a stratified boolean circuit of bounded fan in and bounded fan out. We can suppose that, given $(\mathcal{C}, \mu)$ an instance of $SCVP$, $\mathcal{C}$ is a square boolean circuit whose fan in and fan out is equal to 2, furthermore we can suppose that it is monotone and stratified. Also, Given $(\mathcal{C}, \mu)$ an instance of $MCVP$, we can apply on it the reduction $\mathcal{N}$. We observe that $\mathcal{N}$ is a honest $NC$ reduction, since it is a $\log space$ reduction, and given $(\mathcal{C}, \mu)$ we have that the size of $\mathcal{N}(\mathcal{C}, \mu)$ is bigger than the size of $(\mathcal{C}, \mu)$. Thus, if one wants to prove that $SPP[\mathcal{L}_3]$ is $\sqrt[6]{n}$-hard for $P$, one only has to prove that

$$|\mathcal{N}(\mathcal{C}, \mu)| \leq |(\mathcal{C}, \mu)|^{\frac{6}{2}}$$

Last inequality is straightforward. Thus, we have proven that $SPP[\mathcal{L}_3]$ (and $MC[\mathcal{L}_3]$) is $\sqrt[6]{n}$-hard for $P$. ■

## 8.3 Three dimensional critical avalanches

In this short section we study the statistics of three-dimensional critical avalanches. Furthermore, we analyze the average-case behavior of simulation algorithms solving the problem $GC\left[\mathcal{L}_n^3\right]$.

**Teorema 77** *Let $n$ be a natural number, we have*

1. $L(w_n^n, w_n^n) \in \Omega\left(n^4\right)$.
2. $\Pr_{f,g \in \mathcal{K}(\mathcal{L}_n^3)}\left[L(f,g) \geq \frac{L\left(w_n^1, w_n^1\right)}{2^{72}}\right] \geq \frac{1}{72}$.
3. *There exists a positive constant $K$ such that*

$$\Pr_{f,g \in \mathcal{K}(\mathcal{L}_n^3)}\left[L(f,g) \geq Kn^4\right] \geq \frac{1}{69}$$

**Proof.** Item 2 follows from theorem 54, item 3 follows from items 1 and 2; we prove item 1. Given $\mathcal{L}_n$ a sandpile lattice, we use the symbol $\delta(\mathcal{L}_n)$ to denote the set

$$\left\{w \in V(\mathcal{L}_n)^* : (\{*, w\} \in E(\mathcal{L}_n))\right\}$$

We use the symbol $\delta_n$ to denote the configuration defined by:

$$\text{given } v \in V(\mathcal{L}_n)^*, \ \delta_n(v) = 6 - \deg_{\mathcal{G}_n}(v)$$

We prove that there exists a constant $C$ such that for any $n \geq 1$ we have that $L(w_n, w_n) \geq Cn^4 \in \Omega\left(|\mathcal{L}_n|^{\frac{4}{3}}\right)$. Remember that all the avalanches triggered by $2w_n$ have the same length. Fix $n \geq 2$, we want to lowerbound the length of a very specific avalanche triggered by $2w_n$. Given $n \geq 2$, we can identify the sink of $\mathcal{L}_{n-2}$ with $\delta(\mathcal{L}_n)$ the border of $\mathcal{L}_n$. If we make such an identification, we can think of $\mathcal{L}_{n-2}$ as embedded into $\mathcal{L}_n$, and we can express the configuration $w_n$ as $w_{n-2} + \delta_n + \gamma_n$, where $\gamma_n$ is some configuration on $\mathcal{L}_n$. Note that

$$2w_n = (w_n + \delta_n) + (w_{n-2} + \gamma_n)$$

We know that that

$$\begin{aligned} st_{\mathcal{L}_n}(2w_n) &= st_{\mathcal{L}_n}\left(st_{\mathcal{L}_n}(w_n + \delta_n) + st_{\mathcal{L}_n}(w_{n-2} + \gamma_n)\right) \\ st_{\mathcal{L}_n}(w_n + \delta_n) &= w_n \text{ and } L(w_n, \delta_n) = \left|V(\mathcal{L}_n)^*\right| = (n)^3 \end{aligned}$$

Thus, we have that there exists a configuration $\beta_n$ such that we can pass from the configuration $2w_n$ to the configuration $2w_{n-2} + \beta_n$. Furthermore, we have that the partial avalanche carrying us from $2w_n$ to $2w_{n-2} + \beta_n$ has a length equal to $n^3$. This partial avalanche (it is not a maximal avalanche) is the first stage of the whole stabilization process. In the second stage we work on the subgraph $\mathcal{L}_{n-2}$ with the configuration $2w_{n-2}$. We can claim that after $(n-2)^3$ topplings we can pass from $2w_{n-2}$ to $2w_{n-4} + \beta_{n-1}$. If we continue in this way, going to the core (center) of $\mathcal{L}_n$, we have to generate $\left\lfloor \frac{n}{2} \right\rfloor - 1$ partial avalanches whose lengths are lowerbounded by $n^3, (n-2)^3, ..., \left(n - 2\left(\left\lfloor \frac{n}{2} \right\rfloor - 2\right)\right)^3$ and $\left(n - 2\left(\left\lfloor \frac{n}{2} \right\rfloor - 1\right)\right)^3$ (respectively). Therefore, we have that $L(w_n, w_n) \geq \left( \sum_{i=0}^{\left\lfloor \frac{n}{2} \right\rfloor - 1} (n-2i)^3 \right) \in \Omega\left(n^4\right)$. ■

Let $X_n : \mathcal{K}(\mathcal{L}_n)^2 \to \mathbb{N}$ be the random variable defined by $X_n(f,g) = L(f,g)$.

**Teorema 78** *$E[X_n]$, the expected value of $X_n$, belongs to $\Theta\left(n^4\right)$.*

**Proof.** We know that there exist positive constants $D, K$ such that

1. For all $f, g \in \mathcal{K}(\mathcal{L}_n)$ we have that $X_n(f,g) \leq Dn^4$.
2. $\Pr_{f,g \in \mathcal{K}(\mathcal{L}_n)}\left[X_n(f,g) \geq Kn^4\right] \geq \frac{1}{69}$.

Then, we have that

$$\frac{K}{69} n^4 \leq E[X_n] \leq Dn^4$$

Therefore, we have that $E[X_n] \in \Theta\left(n^4\right) = \Theta\left(|\mathcal{L}_n|^{\frac{4}{3}}\right)$ ■

Now, we come to the analysis of the average-case behavior of simulation algorithms solving the problem $GC[\mathcal{L}_3]$.

**Teorema 79** *$GC[\mathcal{L}_3]$ is $\sqrt[6]{n}$-hard for $P$.*

**Proof.** It follows from the following facts: $SPP[\mathcal{L}_3]$ is $\sqrt[6]{n}$-hard for $P$, and the reduction of $SPP[\mathcal{L}_3]$ in $GC[\mathcal{L}_3]$ is a size-preserving honest $NC$ reduction ■

Next theorem follows easily from the results obtained in section 2, it is the three-dimensional version of theorem 67, it is interesting because it brings together the results concerning the algorithmic hardness of $GC$ and the results concerning the statistics of three-dimensional critical avalanches. Remember that $\mathcal{SA}$ denotes the (naive) sequential sandpile automata simulation algorithm, and remember that the symbol $\mathcal{B}$ denotes the parallel sandpile automata simulation algorithm. We will use the symbol $t^{(3)}_{\mathcal{SA}}(n,f,g)$ to denote the running time of $\mathcal{SA}$ on input $(n,f,g)$, (we define $t^{(3)}_{\mathcal{B}}(n,f,g)$ accordingly).

**Teorema 80** *Let $n \geq 1$ be a natural number.*

1. *There exists a positive constant $K$ such that*

$$\Pr_{f,g\in\mathcal{K}(\mathcal{L}_n)}\left[t_{\mathcal{SA}}(n,f,g) \geq Kn^4\right] \geq \frac{1}{69}$$

2. *There exists a positive constant $K$ such that*

$$\Pr_{f,g\in\mathcal{K}(\mathcal{L}_n)}\left[t_{\mathcal{B}}(n,f,g) \geq Kn\right] \geq \frac{1}{69}$$

**Proof.** The proof is completely analogous to the proof of theorem 67 ■

Theorem 80 suggests that the problem $GC$ is $n^{\frac{1}{6}}$-hard on average, which means that given an algorithm $\mathcal{M}$ computing the problem $GC$, there exists two positive constants $K, D$ such that

$$\Pr_{f,g\in\mathcal{K}(\mathcal{L}_n)}\left[t_{\mathcal{M}}(n,f,g) \geq Kn^{0.5}\right] \geq D$$

**Conjecture 81** *The problem $GC[\mathcal{L}_3]$ is $\sqrt[6]{n}$-hard on average.*

## 8.4 Exercises

1. Let $L$ be a $P$-complete problem. Is there an unbounded function $f_L(n)$, such that $L$ is $f_L(n)$ strict $P$-complete?

2. What can be said about higher dimensional sandpile lattices?

3. U

# 9
# Open problems

## 9.1 Directed sandpiles

We can prove that there are not polynomial bound on avalanche's length, when we consider the class of two-dimensional directed sandpile lattices. This fact rules out the possibility of using the naive *sandpile automata simulation algorithm* to solve, in polynomial time, the problem $SPP$ when restricted to directed sandpiles.

Let $(G, S)$ be a sandpile graph such that $S = \{s\}$ and there exists a path $v_0, v_1, v_2, ..., v_n, s$ with the following two properties:

1. $\deg_+(v_0) = 1$.
2. For any $i$, if $1 \leq i \leq n$, then $\deg_+(v_i) \geq 2$.
3. For any $i \geq 0$ we have that $\deg_-(v_i) = 1$.

**Lemma 82** *Given* $g = \left(|G|^2 + 1\right) e_{v_0}$, *the length of any maximal avalanche triggered by* $g$ *is lowerbounded by* $2^n$.

**Proof.** First we note that, in order to stabilize the sandpile it is necessary to throw at least one chip trough the sink. It implies that $SC_g(v_n) \geq 1$. Note that in order to place one chip on $v_n$ we have to fire $v_{n-1}$ at least one time. Hence, one toppling at node $v_n$ forces at least two topplings at node $v_{n-1}$. Two topplings at node $v_{n-1}$ forces at least four topplings at node $v_{n-2}$, and so on. We can show, using an inductive argument, that one toppling at node $v_n$ forces at least $2^n$ topplings at node $v_0$. Thus, we have proven that the length of any maximal avalanche with initial configuration $\left(|G|^2+1\right)e_{v_0}$ is bigger than $2^n$. ■

**Teorema 83** **??** *There is not polynomial bound on the size of the avalanches for the abelian sandpile model on two-dimensional sandpile directed lattices.*

**Proof.** First at all we define $\left((G_n,S_n)\right)_{n\geq 1}$ a sequence of two-dimensional sandpile directed lattices. Given $n\geq 1$ we define $(G_n,S_n)$ in the following way:

1. $V(G_n)=\{(m,i): m\leq n+1 \text{ and } i\in\{0,1\}\}.$
2. $E(G_n)=A_1\cup A_2\cup A_3$, where
   $A_1=\{((m,0),(m+1,0)): m\leq n\};$
   $A_2=\{((m,1),(m-1,1)): 1\leq m\leq n+1\}$ and
   $A_3$ is equal to
   $\{((m,0),(m,1)): 1\leq m\leq n\}\cup\{((0,1),(0,0)),((n+1,1),(n+1,0))\}.$
3. $S_n=\{(n+1,0)\}.$

Note that the path $(0,0),(1,0),...,(n+1,0)$ satisfies the conditions in the statement of lemma 82, and note that $|G_n|=2n+4$. From lemma 82 we have that the length of any maximal avalanche beginning in $g_n=\left((2n+4)^2+1\right)e_{(0,0)}$ is lowerbounded by $2^n$. ■

**Remark 84** *Note that if we define $g_n$ as $(n+1)e_{(0,0)}$ we obtain the same lower bound on the length of the maximal avalanches triggered by $g_n$.*

Last theorem rules out the possibility of solving in polynomial time the problem $SPP[2]$ by means of the naive sandpile automata simulation algorithm. It does not imply that we can not solve $SPP[2]$ in polynomial time, note that, with some effort we could compute a closed-form formula (of low arithmetical complexity) for the function $h:\mathbb{N}\rightarrow\mathbb{N}^{V(G_n)^*}$ defined by $h(n)=st_{G_n}(g_n)$[1]. So, we can predict, (even better than in polynomial time), the final states of our exponential long avalanches. Can we always predict? At the moment we do not know if $SPP[2]$ belongs to $P$, this problem could be intractable, but we conjecture that $SPP[2]\in P$. Which is the complexity of $SPP[2]$? Is $SPP[2]$ $NP$ complete?

[1]Note that given $n,m\geq 1$ and given $g_{n,m}=me_{(0,0)}\in C(G_n)$, the relaxation of $g_n$ is the configuration $g_n^*$ defined by

$$g_n^*((k,i))=\left\{\begin{array}{c} 1 \text{ if } 1\leq k\leq (m \bmod n+1) \\ 0, \text{ otherwise}\end{array}\right.$$

## 9.2 The complexity of two-dimensional sandpiles

We have proven that $SPP[\mathcal{L}_2]$ is $NC^1$-hard, and we have proven that $SPP[\mathcal{L}_2]$ belongs to $P$. These are the best upper and lower bounds that have been already established for this problem. Also, the gap between upper and lower bounds is still very large. Which is the complexity of $SPP[\mathcal{L}_2]$? We conjecture that $SPP[\mathcal{L}_2]$ is $P$-hard. We consider that establishing tight bounds for the computational complexity of $SPP[\mathcal{L}_2]$ is the most important open problem in the area. It is important to remark that there is some work concerning this problem; Gajardo and Goles [GM] have shown that a proof of $P$ hardness for $SPP[\mathcal{L}_2]$ could not be achieved using Moore's construction. The work of Gajardo-Goles suggests that either $SPP[\mathcal{L}_2]$ is not $P$ complete or the proofs showing that $SPP[\mathcal{L}_2]$ is $P$-hard are far from reach.

There is some weak evidence concerning the possible $P$-completeness of $SPP[\mathcal{L}_2]$, this weak evidence is provided by the work of Schulz [S2]. Let *dist* be the algorithmic problem defined by:

**Problem 85** *(dist; how far is the next critical configuration)*

- *Input:* $(n, f)$, *where* $n \in \mathbb{N}$ *and* $f \in \mathcal{M}(\mathcal{L}_2^n)$.
- *Problem: Compute* $\min_{\|t\|} \{f + t \in \mathcal{K}(\mathcal{L}_2^n)\}$.

Schulz has proven that the problem *dist* is $NP$-complete. Though, there are not a clear connection between the algorithmic hardness of the problems *dist* and $SPP[\mathcal{L}_2]$, we believe that the theorem of Schulz suggests that $SPP[\mathcal{L}_2]$ is $P$-complete.

# References

[Ba] L. Babai. The abelian sandpile model. Manuscript, available at http://people.cs.uchicago.edu/~laci/REU05/.

[BG] L. Babai, I. Gorodezky. Sandpile Transience on the Grid is Polynomially Bounded. Proc. 2007 ACM-SIAM Symposium on Discrete Algorithms, pgs 627-636.

[BTW] P. Bak. *How Nature Works: The Science of Self-organized Criticality.* New York, Copernicus, 1996.

[BL] B. Bjorner, L. Lovasz. Chip Firing Games on Directed Graphs. Journal of Algebaic combinatorics, 1(1991): 305-328.

[C] A. Condon. A Theory of Strict $P$-completeness. STACS(1992):33-44.

[CS] J. Cooper, J. Spencer. Simulating a Random Walk with Constant Error. Combinatory, Probability and Computation. 15(6): 815-822, 2006.

[D] D. Dhar. Theoretical Studies of Self-organized Criticality. *Physica A.* 369:29-70, 2006.

[GGM] A. Gajardo, E. Goles, A. Moreira. Complexity Of Langton´s Ant. Discrete applied mathematics, 117(2002):41-50.

[GG] Gajardo, Goles. *crossing information.*

[GM] Goles, Martinez

[L] C. Langton. Studying Artificial Life with Cellular Automata. Physica D 22(1986):120-149.

[MMG] J. Machta, K. Moriarty, R. Greenlaw. Parallel Algorithm and Dynamic Exponent for Diffusion-limited Agreggation. Physical Review E55, 6211, 1997.

[M] C. Mejia. El modelo de pilas de arena sobre grafos dirigidos y algo de complejidad. Revista integracion: temas de Matematicas, 24(2;2006):101-116.

[MM] C. Mejia, A. Montoya. On the Complexity of Sandpile Prediction Problems. *Electronic Notes in Theoretical Computer Science.* 252:229-245, 2009.

[Mi] P. Miltersen. The computational complexity of one-dimensional sandpiles. *Theory of computing systems.* 41(1):119-125, 2007.

[Mo] C. Moore. Majority-voting Cellular Automata, Ising Dynamics And P-completeness. Journal of statistical physics 88(1997):795-805.

[MN] C. Moore, M. Nilsson. The computational complexity of sandpiles. *Journal of Statistical Physics.* 96:205-224, 1999.

[P] C. Papadimitriou. *Computational Complexity.* Addison wesley, Reading MA, 1994.

[PDDK] V. Priezzhev, D. Dhar, A. Dhar, S. Krishnamurthy. Eulerian Walkers as a Model of Self-organized Criticality. Physical Review Letters, 77:5079-5082, 1996.

[R] W Ruzzo. On Uniform Circuit Complexity. Journal of Computer and Systems Sciences. 22 (1981):365-383.

[S] M. Schulz. On the addition of Recurrent configurations of the Abelian Sandpile Model. ACRI 2008:236-243.

[S2] Schulz, complete problem

[Su] Sudborough

[T] G. Tardos. Polynomial bound for a chip firing game on graphs. *SIAM J. Discrete Mathematics.* 1:397-398, 1988.

[Th] C. Thompson. *Mathematical Statistical Mechanics.* Princenton univ. press, Princenton NY, 1972.

[To] E. Toumpakari. *On the abelian sandpile model.* Ph.D. Thesis, Universidad de Chicago, 2005.

[W] D. Welsh. *Complexity: Knots, Colourings and Counting.* Cambridge University Press, Cambridge (UK), 1993.

[Wo] Wolfram S. *Cellular Automata And Complexity.* Wolfram research, U. champaign IL, 1994.