Time and Space Complexity of Reversible Pebbling*

Richard Kráľovič

Department of Computer Science
Faculty of Mathematics, Physics and Informatics
Comenius University, Bratislava
Slovakia

Abstract. In the context of quantum computing, reversible computations play an important role. In this paper the model of the reversible pebble game introduced by Bennett is considered. Reversible pebble game is an abstraction of a reversible computation, that allows to examine the space and time complexity for various classes of problems. We present a technique for proving lower and upper bounds on time and space complexity. Using this technique we show a partial lower bound on time for optimal space (time for optimal space is not $o(n \lg n)$) and a time-space tradeoff (space $O(\sqrt[k]{n})$ for time $2^k n$) for a chain of length n. Further, we show a tight optimal space bound ($h + \Theta(\lg^* h)$) for a binary tree of height h and we discuss space complexity for a butterfly. By these results we give an evidence, that for reversible computations more resources are needed with respect to standard irreversible computations.

1 Introduction

Standard pebble game was introduced as a graph-theoretic model, that enables to analyse time-space complexity of deterministic computations. In this model, values to be computed are represented by vertices of a directed acyclic graph. An edge from a vertex a to a vertex b represents the fact, that for computing the value a, the value b has to be already known. Computation is modelled by laying and removing pebbles on/from the vertices. Pebbles represent the memory locations. A pebble laying on a certain vertex represents the fact that the value of this vertex is already computed and stored in the memory.

The importance of the pebble game is in the following two step paradigm:

1. the inherent structure of studied problem forms the class of acyclic graphs; investigate time-space complexity of pebbling this class of graphs;
2. apply the obtained time-space results to create a time efficient space restricted computation of the original studied problem.

Various modifications of this game were studied in connection with different models of computations (e.g. pebble game with black and white pebbles for nondeterministic computations, two person pebble game for alternation computations, pebble game with red and blue pebbles for input-output complexity

* Supported in part by grant from VEGA 1/7155/20.

analysis, pebble game with labels for database serializability testing, etc., see [7]).

In connection with quantum computing, the model of reversible computations is very interesting. As the basic laws of quantum physics are reversible, also the quantum computation has to be reversible. That means, that each state of the computation has to uniquely define both the *following* and the *preceding* state of the computation.

Another motivation to examine the model of reversible computation follows from the fact, that reversible operations are not known to require any heat dissipation. With continuing miniaturisation of computing devices, reduction of the energy dissipation becomes very important. Both these reasons for studying reversible computations are mentioned in [9], [2], [5] and [4].

A modification of the standard pebble game for modelling reversible computations is the reversible pebble game. Reversible pebble game enables to analyse time and space complexity and time-space trade-offs of reversible computations.

In this paper, three basic classes of dags are considered: the chain topology, the complete binary tree topology and the butterfly topology. These topologies represent the structure of the most common problems.

It is evident, that minimal space complexity for standard pebble game on chain topology is $O(1)$, minimal time complexity is $O(n)$ and minimal space and time complexities can be achieved simultaneously. For reversible pebble game, in [4] was proved minimal space complexity on the chain topology in the form $O(\lg n)$ and upper bound on time complexity for optimal space complexity in the form $O(n^{\lg 3})$. In [1] it was introduced a pebbling strategy, that yields an upper bound of time-space tradeoff for reversible pebble game on chain in the form: space $O(\frac{k-1}{\lg k} \lg n)$ versus time $\Omega(n^{\frac{\lg(2k-1)}{\lg k}})$.

We show that optimal time for optimal space complexity cannot be $o(n \lg n)$. Further, we show the upper bound on the time-space tradeoff for reversible pebble game on chain in the form: space $O(\sqrt[k]{n})$ versus time $2^k n$.

Minimal space complexity $h+1$ for standard pebble game on a complete binary tree of height h was proved in [6]. We show a tight space bound for reversible pebble game on a complete binary tree in the form $h + \Theta(\lg^* h)$. These results give an evidence, that more resources are needed for reversible computation in comparison with irreversible computation.

2 Preliminaries

Reversible Pebble Game is played on directed acyclic graphs. Let G be a dag. A *configuration on* G is a set of its vertices covered by pebbles. Let C be a configuration, the formula $C(v) = 1$ denotes the fact that the vertex v is covered by a pebble. Analogically, $C(v) = 0$ denotes that the vertex v is uncovered. We denote the number of pebbles used in a configuration C as $\#(C)$. An *empty configuration* on G is denoted as $\mathrm{E}(C)$. Empty configuration is a configuration without pebbles. The rules of *Reversible pebble game* are the following:

R1 A pebble can be laid on a vertex v if and only if all direct predecessors of the vertex v are covered by pebbles.

R2 A pebble can be removed from a vertex v if and only if all direct predecessors of the vertex v are covered by pebbles.

Reversible pebble game differs from standard pebble game in rule R2 – in standard pebble game, pebbles can be removed from any vertex at any time.

An ordered pair of configurations on dag G, such that the second one follows from the first one according to these rules, is called a *transition*.

For our purposes, a transition can be also a pair of two identical configurations. A *nontrivial transition* is a transition not formed by identical configurations.

Important property of a transition in a reversible pebble game is *symmetry*. From the rules of the game follows, that if (C_1, C_2) forms a transition, then also (C_2, C_1) forms a transition.

A *computation on graph* G is a sequence of configurations on G such that each successive pair forms a transition. Let $\mathcal{C}$ be a computation, $\mathcal{C}(i)$ denotes the i-th configuration in the computation $\mathcal{C}$. A computation $\mathcal{C}$ is a *complete computation*, if and only if the first and the last configurations of $\mathcal{C}$ are empty (e.g. $\#(\mathcal{C}(1)) = \#(\mathcal{C}(n)) = 0$, where n is the length of the computation $\mathcal{C}$) and for each vertex v there exists a configuration C in $\mathcal{C}$, such that v is covered in C.

We shall be interested in *space* and *time* complexities of a computation $\mathcal{C}$. *Space* of a computation $\mathcal{C}$ (denoted as $\mathrm{S}(\mathcal{C})$) is the number of pebbles needed to perform the computation – that is the maximum number of pebbles used over all configurations of $\mathcal{C}$. *Time* of a computation $\mathcal{C}$ (denoted as $\mathrm{T}(\mathcal{C})$) is the number of nontrivial transitions in $\mathcal{C}$.

The *minimal space* of the reversible pebble game on the dag G (denoted as $\mathrm{S}_{\min}(G)$) is the minimum of $\mathrm{S}(\mathcal{C})$ over all complete computations $\mathcal{C}$ on G. The *time* $\mathrm{T}(G, s)$ of the reversible pebble game on the dag G with *at most s pebbles* is the minimum of $\mathrm{T}(\mathcal{C})$ over all complete computations $\mathcal{C}$ on G such that $\mathrm{S}(\mathcal{C}) \leq s$.

Let $\mathcal{G}$ be a class of dags. Then the *minimal space function* $\mathrm{S}_{\min}(n)$ of a class $\mathcal{G}$ is the maximum of $\mathrm{S}_{\min}(G)$ over all dags in the subclass $\mathcal{G}_n$. The *time function* $\mathrm{T}(n, s)$ is the maximum of $\mathrm{T}(G, s)$ over all dags G in the subclass $\mathcal{G}_n$.

2.1 Operations on Computations

For proving upper and lower bounds on time and space complexities of the reversible pebble game, it is useful to manipulate formally with reversible computations. We will use an algebraic way to describe computations. An advantage of this approach is in high precision of the description. In this section we introduce some operations for constructing and modifying computations.

For changing state of a particular vertex in a configuration, we use the operation *Put*.

Definition 1. *Let* $G = (V, E)$ *be a dag,* C *be a configuration on* G*. Let* $v \in V$ *and* $h \in \{0, 1\}$*. Then* $\mathrm{Put}(C, v, h)$ *is a configuration on* G *defined as follows:*

– $\mathrm{Put}(C, v, h)(u) = C(u)$, *if* $u \in V$ *and* $u \neq v$;
– $\mathrm{Put}(C, v, h)(u) = h$, *if* $u \in V$ *and* $u = v$.

An important property of reversible computations is the following one: Let G be a dag, G' be a subgraph of G and $\mathcal{C}$ be a computation on G. If we remove all vertices not in G' from all configurations of $\mathcal{C}$, we obtain a reversible computation on G'. The correctness of such construction is clear – we cannot violate any rule of reversible pebble game by removing a vertex from all configurations of a computation. Another important fact is, that removing some configurations from the beginning and the end of a reversible computation does not violate a property of a reversible computation, too.

Also, we can define an operator for a "restriction" of a computation:

Definition 2. *Let* $G = (V, E)$ *be a dag,* $V' \subseteq V$*. Let* $\mathcal{C}$ *be a computation of the length* n *on* G*. A* Restriction $\mathcal{C}' = \mathrm{Rst}(\mathcal{C}, i, j, V')$ *of the computation* $\mathcal{C}$ *to an interval* $\{i \dots j\}$ $(1 \leq i \leq j \leq n)$ *and to a subgraph* $G' = (V', E \cap (V' \times V'))$ *is a computation* $\mathcal{C}'$ *of the length* $j - i + 1$ *on* G' *defined as follows:*

$$(\forall k \in \{1 \dots j - i + 1\})(\forall v \in V')\mathcal{C}'(k)(v) = \mathcal{C}(i + k - 1)(v)$$

We use a notation $\mathrm{Rst}(\mathcal{C}, i, j)$ *when no vertices should be removed (e.g.* $\mathrm{Rst}(\mathcal{C}, i, j) = \mathrm{Rst}(\mathcal{C}, i, j, V)$ *for the graph* $G = (V, E)$*).*

From the symmetry of the rules of the reversible pebble game follows, that reversing a reversible computation does not violate the reversible computation property. We can therefore define an operator *Rev*.

Definition 3. *Let* $\mathcal{C}$ *be a computation on* G *of the length* n*. Then the* reverse *of the computation* $\mathcal{C}$ *(denoted as* $\mathrm{Rev}(\mathcal{C})$*) is a computation on* G *defined as follows:*

$$(\forall i \in \{1 \dots n\})\, \mathrm{Rev}(\mathcal{C})(i) = \mathcal{C}(n + 1 - i)$$

Now we introduce operations, that are inverse to restriction in some sense.

Definition 4. *Let* $\mathcal{C}_1$ *and* $\mathcal{C}_2$ *are computations on a dag* G*, let* $\mathcal{C}_1$ *and* $\mathcal{C}_2$ *have length* n_1 *and* n_2 *respectively. Let* $\mathcal{C}_1(n_1)$ *and* $\mathcal{C}_2(1)$ *form a transition. Then the* join *of computations* C_1 *and* C_2 *(denoted as* $\mathcal{C}_1 + \mathcal{C}_2$*) is a computation on* G *of length* $n_1 + n_2$ *defined as follows:*

– $(\mathcal{C}_1 + \mathcal{C}_2)(i) = \mathcal{C}_1(i)$, *if* $i \leq n_1$
– $(\mathcal{C}_1 + \mathcal{C}_2)(i) = \mathcal{C}_2(i - n_1)$, *if* $i > n_1$

It is clear, that this definition is correct. Configurations $(\mathcal{C}_1 + \mathcal{C}_2)(n_1)$ and $(\mathcal{C}_1 + \mathcal{C}_2)(n_1 + 1)$ form a transition by assumption. All other successive pairs of configurations form transitions, because $\mathcal{C}_1$ and $\mathcal{C}_2$ are computations.

Let C be a configuration on a dag G. Then we can look at a configuration C also as at a computation of length 1, so that $\mathcal{C}(1) = C$. Therefore we can also join a computation with a configuration and vice versa.

The join of two computations is an inverse operation to restriction by removing configurations. Now we define an inverse operation to the restriction performed by removing vertices.

Definition 5. *Let* $G = (V, E)$ *be a dag,* $V_1 \subseteq V$*,* $V_2 \subseteq V$*,* $V_1 \cap V_2 = \emptyset$*. Let* C *be a configuration on the graph* $(V_2, E \cap (V_2 \times V_2))$*. Let* $\{(w, v) | v \in V_1 \wedge w \in V_2 \wedge C(w) = 0 \wedge (w, v) \in E\} = \emptyset$*. Let* $\mathcal{C}$ *be a computation of length* n *on the graph* $(V_1, E \cap (V_1 \times V_1))$*. The* computation $\mathcal{C}$ merged with the configuration C *(denoted as* $\mathcal{C} \cdot C$*) is a computation on the graph* $(V_1 \cup V_2, E \cap ((V_1 \cup V_2) \times (V_1 \cup V_2)))$ *of length* n *defined as follows:*

- $(\mathcal{C} \cdot C)(i)(v) = \mathcal{C}(i)(v)$*, if* $v \in V_1$
- $(\mathcal{C} \cdot C)(i)(v) = C(v)$*, if* $v \in V_2$

This definition is clearly correct. By adding the same configuration to all configurations of some computation $\mathcal{C}$, it is only one way to violate the rules of the reversible pebble game: if some of the added direct predecessors of a vertex, which the pebble is laid on or removed from, are not pebbled. But this is prohibited by the assumption of definition.

Any computation on a graph G can be applied on any graph G' that is isomorphic with G. The *application* of a computation can be defined as follows:

Definition 6. *Let* $\mathcal{C}$ *be a computation of length* n *on a dag* G *and* G' *be a dag isomorphic with* G*. Let* φ *is the isomorphism between* G' *and* G*. Then a computation* $\mathcal{C}$ applied to *the graph* G' *(denoted as* $\mathcal{C}|G'$*) is a computation on* G' *of length* n *such that* $(\mathcal{C}|G')(i)(v) = \mathcal{C}(i)(\varphi(v))$ *for all* $1 \leq i \leq n$ *and for all vertices* v *of* G'*.*

3 Chain Topology

The simplest topology for a pebble game is a *chain*. Chain with n vertices (denoted as $Ch(n)$) is a dag $Ch(n) = (V, E)$, where $V = \{1 \ldots n\}$ and $E = \{(i-1, i) | i \in \{2 \ldots n\}\}$. This topology is an abstraction of a simple straightforward computation, where the result of step $n+1$ can be computed only from the result of step n.

In this section we discuss optimal space complexity for a reversible pebble game on the chain topology – the minimal space function $\mathrm{S}_{\min}(n)$ for Ch, where the subclass Ch_n contains only a chain $Ch(n)$. We will discuss also partial lower and upper bounds for optimal time and space complexities – the time function $\mathrm{T}(n, \mathrm{S}_{\min}(n))$ and the upper bound of the time-space tradeoff for the chain topology.

3.1 Optimal Space for the Chain Topology

For determining space complexity of the reversible pebble game on the chain topology we will examine the maximum length of the chain, that can be pebbled by p pebbles. We denote this length as $\mathrm{S}^{-1}(p)$. It holds, that $\mathrm{S}^{-1}(p) = \max\{m | (\exists \mathcal{C} \in \mathbb{C}_{Ch(m)})\ \mathrm{S}(\mathcal{C}) \leq p\}$, where $\mathbb{C}_{Ch(m)}$ is the set of all computations on the graph $Ch(m)$.

Reversible pebble game on the chain topology was studied in connection with reversible simulation of irreversible computation. C. H. Bennett suggested in [1] a pebbling strategy, whose special case has space complexity $\Theta(\lg n)$. Space optimality of this algorithm was proved in [5] and [4]. This result is formulated in following theorem.

Theorem 1. *It holds that* $\mathrm{S}^{-1}(p) = 2^p - 1$. *Therefore for minimal space function of chain topology* $\mathrm{S}_{\min}(n)$ *it holds*

$$\mathrm{S}_{\min}(n) = \Theta(\lg n)$$

3.2 Optimal Time and Space for the Chain Topology

In this section we present upper and partial lower bounds on time for space optimal reversible pebble game played on a chain topology.

We will use two auxiliary lemmas. Their proofs are not difficult and are left out due to space reasons.

Lemma 1. *Let* $\mathcal{C}$ *be a complete computation of length* l *on* $Ch(n)$, $\mathrm{S}(\mathcal{C}) = \mathrm{S}_{\min}(n)$, $\mathrm{T}(\mathcal{C}) = \mathrm{T}(n, \mathrm{S}_{\min}(n))$. *Let* $i = \min\{i | i \in \{1 \ldots l\} \wedge \mathcal{C}(i)(n) = 1\}$. *Then it holds that*

$$\mathrm{T}(\mathrm{Rst}(\mathcal{C}, 1, i)) = \mathrm{T}(\mathrm{Rst}(\mathcal{C}, i, l)) = \frac{\mathrm{T}(\mathcal{C})}{2}$$

Lemma 2. *Let* $\mathcal{C}$ *be a complete computation on* $Ch(\mathrm{S}^{-1}(p+1))$ *such that* $\mathrm{S}(\mathcal{C}) = p + 1$. *It holds that*

$$\max\{\min\{j | j \in \{1 \ldots n\} \wedge \mathcal{C}(i)(j) = 1\} | i \in \{1 \ldots l\}\} = \mathrm{S}^{-1}(p) + 1$$

Now we prove the upper and partial lower bound on time for space optimal pebble game:

Theorem 2. $\mathrm{T}(\mathrm{S}^{-1}(p+1), p+1) \geq 2\mathrm{S}^{-1}(p) + 2 + 2\mathrm{T}(\mathrm{S}^{-1}(p), p)$

Proof. Let $\mathcal{C}$ be a time optimal complete computation on $Ch(\mathrm{S}^{-1}(p+1))$, such that $\mathrm{S}(\mathcal{C}) = p+1$. Let l be the length of $\mathcal{C}$. Clearly $\mathrm{T}(\mathcal{C}) = \mathrm{T}(\mathrm{S}^{-1}(p+1), p+1)$. We prove, that $\mathrm{T}(\mathcal{C}) \geq 2\mathrm{S}^{-1}(p) + 2 + 2\mathrm{T}(\mathrm{S}^{-1}(p), p)$ holds.

Let $n = \mathrm{S}^{-1}(p)$, $G_1 = (\{n+1\}, \emptyset)$ and G_2 be a graph obtained from $Ch(n)$ by renaming vertices to $n+2 \ldots 2n+1 = \mathrm{S}^{-1}(p+1)$. Let $i = \min\{i | i \in \{1 \ldots l\} \wedge \mathcal{C}(i)(2n+1) = 1\}$. By Lemma 1 it holds $\mathrm{T}(\mathrm{Rst}(\mathcal{C}, 1, i)) = \frac{1}{2}\mathrm{T}(\mathcal{C})$. From Lemma 2 follows, that $\mathrm{Rst}(\mathcal{C}, k, k, \{1 \ldots n+1\}) \neq \mathrm{E}(Ch(n)) \cdot \mathrm{E}(G_1))$ for all k and that there exists j such that $\mathrm{Rst}(\mathcal{C}, j, j, \{1 \ldots n+1\}) = \mathrm{E}(Ch(n)) \cdot \mathrm{Put}(\mathrm{E}(G_1), n+1, 1)$. W.l.o.g. we can assume $j \leq i$ (otherwise we can replace $\mathcal{C}$ by $\mathrm{Rev}(\mathcal{C})$). Let k be a configuration such that $\mathcal{C}(k-1)(n+1) = 0$ and $(\forall q)(k \leq q \leq j)\mathcal{C}(q)(n+1) = 1)$. Clearly $\mathcal{C}(k-1)(n) = \mathcal{C}(k)(n) = 1$.

Now consider the computation $\mathcal{C}_2 = \mathrm{Rst}(\mathcal{C}, 1, k-1, \{1 \ldots n\}) \cdot \mathrm{E}(G_1) \cdot \mathrm{E}(G_2) + \mathrm{Rst}(\mathcal{C}, k, j, \{1 \ldots n\}) \cdot \mathrm{Put}(\mathrm{E}(G_1), n+1, 1) \cdot \mathrm{E}(G_2) + \mathrm{Rst}(\mathcal{C}, 1, i, \{n+2 \ldots 2n+1\}) \cdot$

$\mathrm{E}(Ch(n)) \cdot \mathrm{Put}(\mathrm{E}(G_1), n+1, 1)$. Clearly $\mathrm{S}(\mathcal{C}_2) \leq \mathrm{S}(\mathcal{C})$. Also, $\mathcal{C}_2 + \mathrm{Rev}(\mathcal{C}_2)$ is complete on $Ch(2n+1)$ and $\mathrm{T}(\mathcal{C}_2) \leq \mathrm{T}(\mathrm{Rst}(\mathcal{C}, 1, i))$.

It is clear, that $\mathrm{T}(\mathrm{Rst}(\mathcal{C}, 1, k-1, \{1 \dots n\})) \geq n$ – we cannot pebble n vertices with time less than n. $\mathrm{Rev}(\mathrm{Rst}(\mathcal{C}, k, j, \{1 \dots n\})) + \mathrm{Rst}(\mathcal{C}, k, j, \{1 \dots n\})$ is a space optimal complete computation on $Ch(n)$, therefore $\mathrm{T}(\mathrm{Rst}(\mathcal{C}, k, j, \{1 \dots n\})) \geq \frac{1}{2}\mathrm{T}(n, \mathrm{S}_{\min}(n)) = \frac{1}{2}\mathrm{T}(\mathrm{S}^{-1}(p), p)$. Analogically, $\mathrm{T}(\mathrm{Rst}(\mathcal{C}, 1, i, \{n+2 \dots 2n+1\})) \geq \frac{1}{2}\mathrm{T}(n, \mathrm{S}_{\min}(n)) = \frac{1}{2}\mathrm{T}(\mathrm{S}^{-1}(p), p)$.

From these inequalities follows, that $\mathrm{T}(\mathrm{Rst}(\mathcal{C}, 1, i)) \geq n+1+2\frac{1}{2}\mathrm{T}(n, \mathrm{S}_{\min}(n))$. Therefore $\mathrm{T}(\mathcal{C}) \geq 2n+2+2\mathrm{T}(n, \mathrm{S}_{\min}(n)) = 2\mathrm{S}^{-1}(p) + 2 + 2\mathrm{T}(\mathrm{S}^{-1}(p), p)$. □

Corollary 1. *$\mathrm{T}(n, \mathrm{S}_{\min}(n)) = O(n^{\log_2 3})$, $\mathrm{T}(n, \mathrm{S}_{\min}(n)) \neq o(n \lg n)$,*

Proof. The upper bound was presented in [4]. By solving recurrent inequality proved in preceding theorem, we obtain that $T(n, \mathrm{S}_{\min}(n)) = \Omega(n \lg n)$ for $n = 2^p - 1$. Since this function is a restriction of $T(n, \mathrm{S}_{\min}(n))$ for integer n, function $T(n, \mathrm{S}_{\min}(n))$ cannot be $o(n \lg n)$. □

3.3 Upper Bound on Time-Space Tradeoff for Chain Topology

In the previous section it was analysed time complexity of reversible pebbling for space optimal computations. Now we discuss the time complexity for computations, that are not space optimal.

It is obvious, that for any complete computation $\mathcal{C}$ on $Ch(n)$ it holds $\mathrm{T}(\mathcal{C}) \geq 2n$, because each vertex has to be at least one time pebbled and at least one time unpebbled. It is also easy to see, that space of such computation is exactly n.

Now we will analyse space complexity of complete computations on $Ch(n)$ that are running in time at most $c{\cdot}n$. Let $\mathrm{S}^{-1}(c, k) = \max\{n | \exists \mathcal{C} \in \mathbb{C}_{Ch(n)}\ \mathrm{S}(\mathcal{C}) \leq k \wedge \mathrm{T}(\mathcal{C}) \leq cn\}$, where $\mathbb{C}_{Ch(n)}$ is the set of all complete computations on $Ch(n)$.

Theorem 3. *For a fixed k, it holds $\mathrm{S}^{-1}(2^k, p) = \Omega(p^k)$.*

Proof. We prove a statement $\mathrm{S}^{-1}(2^k, p) \geq c(k)p^k$ by induction on k and p. Let $c(1) = 1$. The base case $\mathrm{S}^{-1}(2^1, p) \geq p$ holds trivially. (It is easy to make a complete computation $\mathcal{C}$ on $Ch(p)$ satisfying $\mathrm{S}(\mathcal{C}) = p$ and $\mathrm{T}(\mathcal{C}) = 2p$.)

Assume by the induction hypothesis that it holds $(\forall k' < k)(\forall p')\mathrm{S}^{-1}(2^{k'}, p') \geq c(k')p'^{k'}$ and $(\forall p' < p)\mathrm{S}^{-1}(2^k, p') \geq c(k)p'^k$. We prove, that $\mathrm{S}^{-1}(2^k, p) \geq c(k)p^k$ holds.

Let $\mathcal{C}_1$ be a complete computation on $Ch(\mathrm{S}^{-1}(2^{k-1}, p-1))$, $\mathrm{S}(\mathcal{C}_1) \leq p-1$, $\mathrm{T}(\mathcal{C}_1) \leq 2^{k-1}\mathrm{S}^{-1}(2^{k-1}, p-1)$. Denote the length of $\mathcal{C}_1$ by l_1. Clearly there exists m such that $\mathcal{C}_1(m)(\mathrm{S}^{-1}(2^{k-1}, p-1)) = 1$.

Let $\mathcal{C}_2$ be a complete computation on $Ch(\mathrm{S}^{-1}(2^k, p-1))$, $\mathrm{S}(\mathcal{C}_2) \leq p-1$, $\mathrm{T}(\mathcal{C}_2) \leq 2^k\mathrm{S}^{-1}(2^k, p-1)$.

Let $G_1 = (\{\mathrm{S}^{-1}(2^{k-1}, p-1)+1\}, \emptyset)$. Let G_2 is a graph obtained from $Ch(\mathrm{S}^{-1}(2^k, p-1))$ by renaming its vertices to $\mathrm{S}^{-1}(2^{k-1}, p-1)+2, \dots, \mathrm{S}^{-1}(2^{k-1}, p-1)+1+\mathrm{S}^{-1}(2^k, p-1)$. Now assume the following computation $\mathcal{C}_3 = \mathrm{Rst}(\mathcal{C}_1, 1, m) \cdot \mathrm{E}(G_1) \cdot \mathrm{E}(G_2) + \mathrm{Rst}(\mathcal{C}_1, m, l_1) \cdot \mathrm{Put}(\mathrm{E}(G_1), \mathrm{S}^{-1}(2^{k-1}, p-1)+1, 1) \cdot \mathrm{E}(G_2) +$

$(\mathcal{C}_2|G_2)\cdot \mathrm{E}(Ch(\mathrm{S}^{-1}(2^{k-1}, p-1)))\cdot \mathrm{Put}(\mathrm{E}(G_1), \mathrm{S}^{-1}(2^{k-1}, p-1)+1, 1)+\mathrm{Rev}(\mathrm{Rst}(\mathcal{C}_1, m, l_1))\cdot \mathrm{Put}(\mathrm{E}(G_1), \mathrm{S}^{-1}(2^{k-1}, p-1)+1, 1)\cdot \mathrm{E}(G_2)+\mathrm{Rev}(\mathrm{Rst}(\mathcal{C}_1, 1, m))\cdot \mathrm{E}(G_1)\cdot \mathrm{E}(G_2)$.

Clearly $\mathcal{C}_3$ is a complete computation on $Ch(\mathrm{S}^{-1}(2^{k-1}, p-1) + 1 + \mathrm{S}^{-1}(2^k, p-1))$ satisfying $\mathrm{S}(\mathcal{C}_3) \leq p$ and $\mathrm{T}(\mathcal{C}_3) \leq 2\mathrm{T}(\mathcal{C}_1) + 2 + \mathrm{T}(\mathcal{C}_2) \leq 2^k \mathrm{S}^{-1}(2^{k-1}, p-1) + 2 + 2^k \mathrm{S}^{-1}(2^k, p-1) \leq 2^k(\mathrm{S}^{-1}(2^{k-1}, p-1) + 1 + \mathrm{S}^{-1}(2^k, p-1))$. Therefore $\mathrm{S}^{-1}(2^k, p) \geq \mathrm{S}^{-1}(2^{k-1}, p-1) + 1 + \mathrm{S}^{-1}(2^k, p-1)$.

By induction hypothesis we have $\mathrm{S}^{-1}(2^k, p) \geq c(k-1)(p-1)^{k-1} + c(k)(p-1)^k$. For a suitable value of $c(k)$ (we can choose $c(k) = \frac{c(k-1)}{2^k}$) it holds that $c(k-1)(p-1)^{k-1} + c(k)(p-1)^k \geq c(k)p^k$. Also there exists $c(k)$ such that $\mathrm{S}^{-1}(2^k, p) \geq c(k)p^k$. □

Corollary 2. *Let k be fixed. Then $O(\sqrt[k]{n})$ pebbles are sufficient for a complete computation on $Ch(n)$ with time $O(2^k n)$.*

Another upper bound of the time-space tradeoff for the reversible pebbling on chain topology can be obtained by using Bennett's pebbling strategy introduced in [1]. Since this strategy pebbles chain of length k^n with $n(k-1)+1$ pebbles in time $(2k-1)^n$, it yields time-space tradeoff in the form: space $O(\frac{k-1}{\lg k} \lg n)$ versus time $\Omega(n^{\frac{\lg(2k-1)}{\lg k}})$.

4 Binary Tree Topology

In this section we will discuss space complexity of reversible pebble game on complete binary trees. A *complete binary tree* of height 1 (denoted as $\mathrm{Bt}(1)$) is a graph containing one vertex and no edges. A complete binary tree of height $h > 1$ (denoted as $\mathrm{Bt}(h)$) consists of a root vertex and two subtrees, that are complete binary trees of height $h-1$.

This topology represents a class of problems, where the result can be computed from two different subproblems.

We denote the root vertex of $Bt(h)$ as $\mathrm{R}(Bt(h))$, the left subtree of $Bt(h)$ as $\mathrm{Lt}(Bt(h))$ and the right subtree of $Bt(h)$ as $\mathrm{Rt}(Bt(h))$.

As mentioned in section 2, we denote the minimal number of pebbles needed to perform a complete computation on $\mathrm{Bt}(h)$ as $\mathrm{S}_{\min}(h)$. In the sequel we also consider the minimal number of pebbles needed to perform a computation from the empty configuration to a configuration, where only the root is pebbled.

Definition 7. *Let $\mathcal{C}$ be a computation of length l on* $\mathrm{Bt}(h)$. *Let $\mathcal{C}(1) = \mathrm{E}(\mathrm{Bt}(h))$ and $\mathcal{C}(l) = \mathrm{Put}(\mathrm{E}(\mathrm{Bt}(h)), \mathrm{R}(\mathrm{Bt}(h)), 1)$. Then $\mathcal{C}$ is called a* semicomplete computation.

The minimal number of pebbles needed to perform a semicomplete computation on $\mathrm{Bt}(h)$ (e.g. $\min\{\mathrm{S}(\mathcal{C})\}$, where $\mathcal{C}$ is a semicomplete computation) will be denoted as $\mathrm{S}'_{\min}(h)$.

We will use the following inequalities between $\mathrm{S}_{\min}(h)$ and $\mathrm{S}'_{\min}(h)$. Their proofs are not difficult and are left out due to space reasons.

Lemma 3. $\mathrm{S}_{\min}(h)+1 \geq \mathrm{S}'_{\min}(h) \geq \mathrm{S}_{\min}(h)$

Lemma 4. $\mathrm{S}'_{\min}(h+1) = \mathrm{S}_{\min}(h)+2$

4.1 Tight Space Bound for Binary Tree Topology

From the previous lemmas follows, that $\mathrm{S}'_{\min}(h)$ equals to h plus the number of such $i < h$, that $\mathrm{S}'_{\min}(i) = \mathrm{S}_{\min}(i)$. In the following considerations we use a function ${\mathrm{S}'}^{-1}_{\min}$. The value $h = {\mathrm{S}'}^{-1}_{\min}(p)$ denotes the maximal height of binary tree that can be pebbled by a semicomplete computation, that uses at most $h+p$ pebbles. Formally, ${\mathrm{S}'}^{-1}_{\min}(p) = \max\{h|\exists \mathcal{C} \in \mathrm{Sc}_h \wedge \mathrm{S}(\mathcal{C}) = h+p\}$, where Sc_h is the set of all semicomplete computations on $\mathrm{Bt}(h)$. From the definition of ${\mathrm{S}'}^{-1}_{\min}(p)$ follows, that $\mathrm{S}'_{\min}(h) = h + ({\mathrm{S}'}^{-1}_{\min})^{-1}(h)$.

Now we prove the upper (lower) bound of ${\mathrm{S}'}^{-1}_{\min}(p)$. From that follows lower (upper) bound of $\mathrm{S}'_{\min}(h)$ and therefore also lower (upper) bound of $\mathrm{S}_{\min}(h)$ respectively.

Lemma 5. *Let* $h' = {\mathrm{S}'}^{-1}_{\min}(p)$, $h = {\mathrm{S}'}^{-1}_{\min}(p+1)$. *Then the following inequality holds:*

$$h - h' - 1 \leq 2^{h'+p+1} - 1$$

Proof. A configuration on a binary tree is called *opened*, if there exists a path from the root to some leaf of the tree, such that no pebble is laid on this path. Otherwise, the configuration is called *closed*.

From the assumption $h = {\mathrm{S}'}^{-1}_{\min}(p+1)$ follows, that there exists some semicomplete computation $\mathcal{C}$ of length l on $\mathrm{Bt}(h)$, such that $\mathrm{S}(\mathcal{C}) = h+p+1$. Let i be the first configuration of $\mathcal{C}$, such that $\mathcal{C}(j)(\mathrm{R}(\mathrm{Bt}(h))) = 1$ for any $j \geq i$ (e.g. $i = \min\{i|(\forall j \geq i)\ \mathcal{C}(j)(\mathrm{R}(\mathrm{Bt}(h))) = 1\}$).

Because $\mathcal{C}$ is a reversible computation, $\mathcal{C}(i)(\mathrm{R}(\mathrm{Lt}(\mathrm{Bt}(h)))) = \mathcal{C}(i)(\mathrm{R}(\mathrm{Rt}(\mathrm{Bt}(h)))) = 1$. Therefore $\mathrm{Put}(\mathcal{C}(i), \mathrm{R}(\mathrm{Bt}(h)), 0)$ is a closed configuration. Because $\mathrm{Put}(\mathcal{C}(l), \mathrm{R}(\mathrm{Bt}(h)), 0) = \mathrm{E}(\mathrm{Bt}(h))$, this configuration is opened. Let j be the minimal number such that $j \geq i$ and $\mathrm{Put}(\mathcal{C}(j), \mathrm{R}(\mathrm{Bt}(h)), 0)$ is opened.

Because $\mathrm{Put}(\mathcal{C}(j), \mathrm{R}(\mathrm{Bt}(h)), 0)$ is opened and $\mathrm{Put}(\mathcal{C}(j-1), \mathrm{R}(\mathrm{Bt}(h)), 0)$ is closed and $\mathcal{C}$ is a reversible computation, there exists exactly one path in $\mathcal{C}(j)$ from the root to a leaf, such that no pebble is laid on it. Without loss of generality we can assume, that this path is $\mathrm{R}(\mathrm{Bt}(h)), \mathrm{R}(\mathrm{Rt}(\mathrm{Bt}(h))), \mathrm{R}(\mathrm{Rt}^2(\mathrm{Bt}(h))), \ldots, \mathrm{R}(\mathrm{Rt}^{h-1}(\mathrm{Bt}(h)))$.

Now we prove, that for each k, $h \geq k \geq h'+2$ and for each p, $i \leq p < j$, it holds that $\#(\mathcal{C}(p)(\mathrm{Lt}(\mathrm{Rt}^{h-k}(\mathrm{Bt}(h))))) > 0$.

Assume, that this conjecture does not hold. Let k be the maximal number such that violates this conjecture. Let p be the maximal number such that $i \leq p < j$ and $\#(\mathcal{C}(p)(\mathrm{Lt}(\mathrm{Rt}^{h-k}(\mathrm{Bt}(h))))) = 0 \vee \#(\mathcal{C}(p)(\mathrm{Rt}(\mathrm{Rt}^{h-k}(\mathrm{Bt}(h))))) = 0$. Without loss of generality, let $\#(\mathcal{C}(p)(\mathrm{Lt}(\mathrm{Rt}^{h-k}(\mathrm{Bt}(h))))) = 0$. Because $\mathrm{Put}(\mathcal{C}(p), \mathrm{R}(\mathrm{Bt}(h)), 0)$ is closed, in a configuration $\mathcal{C}(p)$ is pebbled at least one vertex from $\mathrm{R}(\mathrm{Rt}(\mathrm{Bt}(h)))$, $\mathrm{R}(\mathrm{Rt}^2(\mathrm{Bt}(h)))$, $\ldots$, $\mathrm{R}(\mathrm{Rt}^{h-k}(\mathrm{Bt}(h)))$. In a configuration $\mathcal{C}(j)$ are all these vertices unpebbled. Let q be the minimal number such that $q > p$ and all these vertices are unpebbled in $\mathcal{C}(q)$. Because $\mathcal{C}$ is a reversible computation, $\mathcal{C}(q-1)(\mathrm{R}(\mathrm{Rt}^{h-k}(\mathrm{Bt}(h)))) = 1$, $\mathcal{C}(q)(\mathrm{R}(\mathrm{Rt}^{h-k}(\mathrm{Bt}(h)))) = 0$ and $\mathcal{C}(q)(\mathrm{R}(\mathrm{Lt}(\mathrm{Rt}^{h-k}(\mathrm{Bt}(h))))) = 1$. Now consider the computation $\mathcal{C}' = \mathrm{Rst}(\mathcal{C}, p, q, \mathrm{Lt}(\mathrm{Rt}^{h-k}(\mathrm{Bt}(h))))$. Computation $\mathcal{C}' + \mathrm{Rev}(\mathcal{C}')$ is a complete computation on $\mathrm{Lt}(\mathrm{Rt}^{h-k}(\mathrm{Bt}(h)))$ (this graph is isomorphic to $\mathrm{Bt}(k-1)$). Space of this computation is at most $\mathrm{S}(\mathcal{C}' + \mathrm{Rev}(\mathcal{C}')) \leq \mathrm{S}(\mathcal{C}) - (3 + h - k) = k + p - 2$. From our assumption follows, that space for any semicomplete computation on $\mathrm{Bt}(k-1)$ is at least $k+p$. From Lemma 3 follows, that the space for any complete computation on $\mathrm{Bt}(k-1)$ is at least $k + p - 1$, what is a contradiction.

Now consider $\mathcal{C}_2 = \mathrm{Rst}(\mathcal{C}, i, j, \mathrm{R}(\mathrm{Rt}(\mathrm{Bt}(h))) \cup \mathrm{R}(\mathrm{Rt}^2(\mathrm{Bt}(h))) \cup \ldots \cup \mathrm{R}(\mathrm{Rt}^{h-h'-1}(\mathrm{Bt}(h))))$. It is a computation on a graph isomorphic to $Ch(h-h'-1)$. In the first configuration of $\mathcal{C}_2$, vertex $\mathrm{R}(\mathrm{Rt}(\mathrm{Bt}(h)))$ is pebbled. In the last configuration of $\mathcal{C}_2$, no vertex is pebbled. Therefore $\mathrm{Rev}(\mathcal{C}_2) + \mathcal{C}_2$ is a complete computation on a graph isomorphic to $Ch(h - h' - 1)$.

Because for each k, $h \geq k \geq h' + 2$ and for each p, $i \leq p \leq j$, it holds that $\#(\mathcal{C}(p)(\mathrm{Lt}(\mathrm{Rt}^{h-k}(\mathrm{Bt}(h))))) > 0$ and $\mathcal{C}(p)(\mathrm{R}(\mathrm{Bt}(h))) = 1$, we can estimate upper bound for space of $\mathcal{C}_2$: $\mathrm{S}(\mathcal{C}_2) \leq (h+p+1)-(1+h-h'-1) = h'+p+1$. Using space upper bound for chain topology (Theorem 1) we have $h - h' - 1 \leq 2^{h'+p+1} - 1$. □

Lemma 6. *Let* $h' = \mathrm{S'}^{-1}_{\min}(p)$*,* $h = \mathrm{S'}^{-1}_{\min}(p+1)$*. Then the following inequality holds:*

$$h - h' - 1 \geq 2^{h'+p-2} - 1$$

Proof. We prove by induction, that for each $k \in \{h' + 1 \ldots h' + 2^{h'+p-2}\}$ there exists a semicomplete computation $\mathcal{C}$ on $Bt(k)$ such that $\mathrm{S}(\mathcal{C}) \leq k + p + 1$. This implies, that $h \geq h' + 2^{h'+p-2}$.

The base case is $k = h' + 1$. By assumption there exists a semicomplete computation $\mathcal{C}$ on $\mathrm{Bt}(h')$ such that $\mathrm{S}(\mathcal{C}) = h' + p$. After applying $\mathcal{C}$ to $\mathrm{Lt}(\mathrm{Bt}(k))$ and $\mathrm{Rt}(\mathrm{Bt}(k))$, pebbling $\mathrm{R}(\mathrm{Bt}(k))$ and applying reversed $\mathcal{C}$ to $\mathrm{Lt}(\mathrm{Bt}(k))$ and $\mathrm{Rt}(\mathrm{Bt}(k))$ we obtain a semicomplete computation on $\mathrm{Bt}(k)$ that uses at most $h' + p + 2 = k + p + 1$ pebbles.

Now assume that the induction hypothesis holds for each $i \in \{h'+1 \ldots k-1\}$. We construct a computation $\mathcal{C}$ on $\mathrm{Bt}(k)$ as follows: At first we apply semicomplete computations on $\mathrm{Lt}(\mathrm{Bt}(k))$, $\mathrm{Lt}(\mathrm{Rt}(\mathrm{Bt}(k)))$, $\ldots$, $\mathrm{Lt}(\mathrm{Rt}^{k-h'-2}(\mathrm{Bt}(k)))$, $\mathrm{Lt}(\mathrm{Rt}^{k-h'-1}(\mathrm{Bt}(k)))$, $\mathrm{Rt}^{k-h'}(\mathrm{Bt}(k))$ sequentially. By induction hypothesis, the space of a semicomplete computation on $\mathrm{Lt}(\mathrm{Rt}^i(\mathrm{Bt}(k)))$ is less than or equal to $(k-i-1)+p+1$ for $i \leq k - h' - 2$. By assumption, the space of a semicomplete computation on $\mathrm{Lt}(\mathrm{Rt}^{k-h'-1}(\mathrm{Bt}(k)))$ and $\mathrm{Rt}(\mathrm{Rt}^{k-h'-1}(\mathrm{Bt}(k)))$ is less than or equal to $h' + p$. Therefore space of this part of $\mathcal{C}$ is less than or equal to $k + p$.

In the second part of $\mathcal{C}$, we perform a space optimal complete computation on a chain consisting of vertices $\mathrm{R}(\mathrm{Bt}(k))$, $\mathrm{R}(\mathrm{Rt}(\mathrm{Bt}(k)))$, $\ldots$, $\mathrm{R}(\mathrm{Rt}^{k-h'-1}(\mathrm{Bt}(k)))$. Due to the Theorem 1, space of this part is less than or equal to $\lceil \log_2(k-h'+1) \rceil + k-h'+1$. Because $k \leq h' + 2^{h'+p-2}$, it holds $\lceil \log_2(k-h'+1) \rceil + k-h'+1 \leq k+p$.

The third part of the computation $\mathcal{C}$ is the reversed first part.

Also, $\mathcal{C}$ is a complete computation on $\mathrm{Bt}(k)$ and $\mathrm{S}(\mathcal{C}) \leq k + p$. Hence, $\mathrm{S}_{\min}(k) \leq k + p$. Using Lemma 3, $\mathrm{S}'_{\min}(k) \leq k + p + 1$. Therefore there exists a semicomplete computation on $\mathrm{Bt}(k)$ with space less than $k + p + 1$. □

Lemma 7. *For $p \geq 2$ it holds that* $2^{\mathrm{S'}^{-1}_{\min}(p)} \leq \mathrm{S'}^{-1}_{\min}(p+1) \leq 2^{4\,\mathrm{S'}^{-1}_{\min}(p)}$.

Proof. Let $h' = \mathrm{S'}^{-1}_{\min}(p)$, $h = \mathrm{S'}^{-1}_{\min}(p+1)$. From Lemma 5 follows $h - h' \leq 2^{h'+p+1}$, what is equivalent to $\mathrm{S'}^{-1}_{\min}(p+1) \leq 2^{\mathrm{S'}^{-1}_{\min}(p)+p+1} + \mathrm{S'}^{-1}_{\min}(p)$.

From the definition of $\mathrm{S'}^{-1}_{\min}$ and from Lemma 3 and Lemma 4 trivially follows, that $\mathrm{S'}^{-1}_{\min}(p) \geq 2p$. Therefore $2^{\mathrm{S'}^{-1}_{\min}(p)+p+1} + \mathrm{S'}^{-1}_{\min}(p) \leq 2^{4\,\mathrm{S'}^{-1}_{\min}(p)}$ for $p \geq 1$. Also, the second inequality holds.

From Lemma 6 follows $h - h' \geq 2^{h'+p-2}$, what is equivalent to $\mathrm{S'}^{-1}_{\min}(p+1) \geq 2^{\mathrm{S'}^{-1}_{\min}(p)+p-2} + \mathrm{S'}^{-1}_{\min}(p)$. Therefore for $p \geq 2$ it holds $\mathrm{S'}^{-1}_{\min}(p+1) \geq 2^{\mathrm{S'}^{-1}_{\min}(p)}$. □

Theorem 4. $\mathrm{S}_{\min}(h) = h + \Theta(\lg^*(h))$

Proof. From the previous lemma follows, that $\mathrm{S'}^{-1}_{\min}(p) = O(\overbrace{16^{16^{\cdots 16}}}^{p})$ and that $\mathrm{S'}^{-1}_{\min}(p) = \Omega(\overbrace{2^{2^{\cdots 2}}}^{p})$. Because $\mathrm{S}'_{\min}(h) = h + (\mathrm{S'}^{-1}_{\min})^{-1}(h)$, it holds that $\mathrm{S}'_{\min}(h) = h + \Omega(\lg^*(h))$ and $\mathrm{S}'_{\min}(h) = h + O(\lg^*(h))$. Therefore $\mathrm{S}'_{\min}(h) = h + \Theta(\lg^*(h))$. From Lemma 3 follows, that $\mathrm{S}_{\min}(h) = h + \Theta(\lg^*(h))$. □

4.2 Extension to Butterflies

Butterfly graphs create important class of graphs to study, as they share superconcentrator property and the butterflies form inherent structure of some important problems in numerical computations, as discrete FFT.

A butterfly graph of order d is a graph $G = (V, E)$, where $V = \{1 \ldots d\} \times \{0 \ldots 2^{d-1} - 1\}$ and $E = \{((i,j),(i+1, j \operatorname{xor} 2^{i-1})) | 1 \leq i < d, 0 \leq j \leq 2^{d-1} - 1\}$. This graph can be decomposed into 2^{d-1} complete binary trees of height d. The root of i-th tree is vertex $(1, i)$ and this tree contains all vertices, that can be reached from the root.

The decomposition property implies, that the minimal space complexity of a complete computation on butterfly graph of order d cannot be lower than the minimal space complexity on a complete binary tree of height d (otherwise we can restrict a complete computation on butterfly to any binary tree to obtain a contradiction).

On the other side, by sequentially applying complete computations to all binary trees obtained by decomposition of the butterfly graph, we obtain a complete computation on it. Also, we can construct a complete computation on a butterfly graph of order d with space complexity equal to minimal space complexity of the binary tree of height d. Therefore the minimal space complexity of the butterfly topology equals to the minimal space complexity of the binary tree topology (e.g. the minimal space for a butterfly graph of order d is $d + \Theta(\lg^*(d))$).

5 Conclusion

In this paper we have analysed an abstract model for reversible computations – a reversible pebble game. We have described a technique for proving time and space complexity bounds for this game and presented a tight optimal space bound for a chain topology, upper and partial lower bounds on time of optimal space for a chain topology, an upper bound on time-space tradeoff for a chain topology and a tight optimal space bound for a binary tree topology. These results implies, that reversible computations require more resources than standard irreversible computations. (For a space complexity of a chain topology it is $\Theta(1)$ vs. $\Theta(\lg n)$ and for a space complexity of a binary tree topology it is $h + \Theta(\log^*(h))$ vs. $h + \Theta(1)$.)

For further research, it would be interesting to examine the time complexity of the reversible pebble game for tree and butterfly topology and to consider other important topologies, for example pyramids.

Acknowledgements

I would like to thank Peter Ružička for his valuable advice and consultations.

References

1. Bennett, C. H.: Time-space trade-offs for reversible computation. SIAM J. Comput., 18(1989), pp. 766–776
2. Buhrman, H., Tromp J. and Vitányi, P.: Time and space bounds for reversible simulation. Proc. ICALP 2001, LNCS 2076, Springer-Verlag, 2001
3. Levine, R. Y. and Sherman, A. T.: A note on Bennett's time-space tradeoff for reversible computation. SIAM J. Computing, 19(4):673–677, 1990
4. Li, M. and Vitányi P. M. B.: Reversibility and adiabatic computation: trading time and space for energy. Proceedings of the Royal Society of London Ser. A, 452:1–21, 1996.
5. Li, M. and Vitányi P. M. B.: Reversible simulation of irreversible computation. Proc. 11th IEEE Conference on Computational Complexity, Philadelphia, Pennsylvania, May 24–27, 1996
6. Paterson, M. S. and Hewitt, C. E.: Comparative Schematology, MAC Conf. on Concurrent Systems and Parallel Computation, 1970, pp. 119–127
7. Ružička, P.: Pebbling – The Technique for Analyzing Computation Efficiency. SOFSEM'89, 1989, pp. 205–224
8. Ružička, P. and Waczulík, J.: On Time-Space Trade-Offs in Dynamic Graph Pebbling. MFCS'93, LNCS 711, Springer-Verlag, 1993, pp. 671–681
9. Williams, R.: Space-Efficient Reversible Simulations, DIMACS REU report, July 2000
10. Zavarský, A.: On the Cost of Reversible Computations: Time-Space Bounds on Reversible Pebbling. Manuscript, 1998