Isoperimetric and Isodiametric Functions of Groups

Mark V. Sapir, Jean-Camille Birget, Eliyahu Rips*

May 19, 1997

Abstract

This is the first of two papers devoted to connections between asymptotic functions of groups and computational complexity. One of the main results of this paper states that if for every m the first m digits of a real number $\alpha \geq 4$ are computable in time $\leq C2^{2^{Cm}}$ for some constant $C > 0$ then n^α is equivalent ("big O") to the Dehn function of a finitely presented group. The smallest isodiametric function of this group is $n^{3/4\alpha}$. On the other hand if n^α is equivalent to the Dehn function of a finitely presented group then the first m digits of α are computable in time $\leq C2^{2^{2^{Cm}}}$ for some constant C. This implies that, say, functions $n^{\pi+1}$, n^{e^2} and n^α for all rational numbers $\alpha \geq 4$ are equivalent to the Dehn functions of some finitely presented group and that n^π and n^α for all rational numbers $\alpha \geq 3$ are equivalent to the smallest isodiametric functions of finitely presented groups.

Moreover we describe all Dehn functions of finitely presented groups $\succ n^4$ as time functions of Turing machines modulo two conjectures:

1. Every Dehn function is equivalent to a superadditive function.
2. The square root of the time function of a Turing machine is equivalent to the time function of a Turing machine.

Contents

*The research of the first two authors was supported in part by NSF grants DMS 9623284, DMS 9203981.

1 Preliminaries

A function $f : \mathbf{N} \to \mathbf{N}$ is called an *isoperimetric function* of a finite presentation $\mathcal{P} = \langle X \mid R \rangle$ of a group G if for every number n and every word w over X which is equal to 1 in G, $|w| \leq n$, there exists a van Kampen diagram over $\mathcal{P}$ whose boundary label is w and area $\leq f(n)$; in other words, w is a product of at most $f(n)$ conjugates of the relators from R (see [21], [9], [13], [3]).

A function is called an isodiametric function of a finite presentation $\mathcal{P} = \langle X \mid R \rangle$ if for every number n and every word w over X which is equal to 1 in G, $|w| \leq n$, there exists a van Kampen diagram over $\mathcal{P}$ whose boundary label is w and diameter $\leq f(n)$; in other words, w is a product of conjugates $x_i^{-1} r_i x_i$ of the relators r_i from R and the lengths of the words x_i are bounded by $f(n)$ (see [8], [7], [11]).

The smallest isoperimetric function of a finite presentation $\mathcal{P}$ is called the Dehn function of $\mathcal{P}$.

Let $f, g : \mathbf{N} \to \mathbf{N}$ be two functions. We write $f \preceq g$ if there exist non-negative constants a, b, c, d such that $f(n) \leq ag(bn) + cn + d$. All functions $g(n)$ which we are considering in this paper grow at least as fast as n. In this case $f(n) \preceq g(n)$ if and only if $f(n) \leq ag(bn)$ for some positive constants a, b. Two functions f, g are called *equivalent* if $f \preceq g$ and $g \preceq f$.

It is well known that Dehn functions and the smallest isodiametric functions corresponding to different finite presentations of the same group are equivalent (see [21] or [2], [11]). This allows us to speak about *the Dehn function* and the smallest isdiametric function of a finitely presented group. *Sometimes we shall also say that a function $f(n)$ is the Dehn function (the smallest isodiametric function) of group G when in fact $f(x)$ is only equivalent to the Dehn function (the smallest isodiametric function) of a presentation of this group.*

We shall not distinguish between equivalent functions in this paper. Thus we do not distinguish, say, functions 2^n and 3^n, n^α and $[n^\alpha]$, 1 and $2n$. But we distinguish n^α and

n^β, where α and β are different numbers greater than 1. We also distinguish, say n^α and $n^\alpha \log n$ for $\alpha \geq 1$.

It is known [3] that every function n^k where $k \geq 1$ is an integer is the Dehn function of some finitely presented group. Some other functions like 2^n are also Dehn functions. Some information about the class of Dehn functions can be found in [21], [12], [2], [9], [13], [10] and other papers. It is also known that not every reasonable function is the Dehn function of some finitely presented group. For example, as was shown by Gromov [12], there are no Dehn functions which are strictly between n and n^2. Thus there exists a gap between n and n^2. Information about isodiametric functions can be found in [8, 11, 7].

Several people including Hamish Short asked whether there are other gaps. Answering another question of Short, M. Bridson proved that there are finitely presented groups with Dehn functions equivalent n^α for some non-integral rational numbers α [6].

It is worth mentioning that a related class of *growth functions* of finitely generated groups contains many gaps. For example, from a well known theorem by Gromov [13] if the growth function of a finitely generated group G is bounded from above by a polynomial then G has a nilpotent subgroup of finite index. This easily implies that its growth function is equivalent to n^k for some natural number k. Thus there does not exist a finitely generated group with growth function equivalent to, say, $n^{4.5}$ or $n^{\pi+1}$.

In this paper, we shall prove that for every relatively fast computable number $\alpha \geq 4$ there exists a finitely presented group with Dehn function equivalent to n^α and the smallest isodiametric function equivalent to $n^{3\alpha/4}$. On the other hand we shall prove that if n^α is equivalent to the Dehn function of a finitely presented group then α is a relatively fast computable number (see Corollary 1.4). For example, n^α is a Dehn function of a finitely presented group if $\alpha \geq 4$ is rational or if $\alpha = \pi + 1$, etc.

We show that there exists a close relationship between Dehn functions and complexity functions of Turing machines. This paper is the first of two papers where we explore this relationship, the next paper will be joint with A. Yu. Ol'shanskii. One of the main results of the present paper (Theorem 1.1) says that every Dehn function is a time function of some (not necessarily deterministic) Turing machine. The class of time functions of Turing machines is very large. It contains all relatively fast computable functions (see below). So it is very surprising that, as the other result of this paper states, there is virtually no difference between the class of Dehn functions $\succ n^4$ and the class of time functions $\succ n^4$. As an immediate corollary we get that the class of Dehn functions is very large. For example all functions of the form $n^\alpha (\log n)^\beta$ for $\alpha > 4$ are Dehn functions of finitely presented groups.

There is a conjecture that every isodiametric function is the space function of some Turing machine. Notice that it is not known that the set of space functions $> n$ differs from the set of time functions $> n$.

In order to formulate the main results of our paper, we need some more definitions.

Let M be an arbitrary (deterministic or nondeterministic) Turing machine. For every natural number n let $T(n)$ be the smallest number such that for every acceptable word w with $|w| \leq n$ there exists a computation of length $\leq T(n)$ which accepts w. Notice that since M may be nondeterministic, there could be several accepting computations for the same initial configuration. The function $T(n)$ is called the *time function* of the Turing machine M.

We call a function f *superadditive* if for all natural numbers m, n we have $f(m+n) \geq f(m) + f(n)$. We do not know any Dehn function of a finitely presented group which is not equivalent to a superadditive function and Sapir conjectures that there are no such Dehn functions. In [15] Guba and Sapir proved that every free product $G * H$ where G and H are non-trivial groups, has a superadditive Dehn function. For example, for every group G the free product $G * \mathbf{Z}$ has a superadditive Dehn function (here $\mathbf{Z}$ is the infinite cyclic group).

The main results of our paper are the following.

Theorem 1.1 *Every Dehn function of a finitely presented group is equivalent to the time function of some (not necessarily deterministic) Turing machine.*

Theorem 1.2 *Let $\mathcal{D}_4$ be the set of all Dehn functions $d(n) \geq n^4$ of finitely presented groups. Let $\mathcal{T}_4$ be the set of time functions $t(n) \geq n^4$ of arbitrary Turing machines* [1]*. Let $\mathcal{T}^4$ be the set of superadditive functions which are fourth powers of time functions. Then*

$$\mathcal{T}^4 \subseteq \mathcal{D}_4 \subseteq \mathcal{T}_4.$$

This theorem is a corollary of Theorem 1.1 and the following result.

Theorem 1.3 *Let $L \subseteq X^+$ be a language accepted by a Turing machine M with a time function $T(n)$ for which $T(n)^4$ is superadditive. Then there exists a finitely presented group $G(M) = \langle A \rangle$ with Dehn function equivalent to $T(n)^4$, the smallest isodiametric function equivalent to $T^3(n)$, and there exists an injective map $K : X^+ \to (A \cup A^{-1})^+$ such that*

1. *$|K(u)| = O(|u|)$ for every $u \in X^+$;*
2. *$u \in L$ if and only if $K(u) = 1$ in G;*
3. *$K(u)$ is computable in time $O(|u|)$ by a deterministic Turing machine.*

It is not known whether $\mathcal{T}^4$ coincides (up to equivalence) with the set of superadditive functions in $\mathcal{T}_4$. It is quite possible that these sets coincide. If this were true and in addition all Dehn function were superadditive (we have discussed this conjecture above) then $\mathcal{D}_4$ would be equal to the set of all superadditive functions from $\mathcal{T}_4$. This is why we said before that the class of Dehn functions $\succ n^4$ and the class of time functions $\succ n^4$ virtually coincide.

Using Theorem 1.3 and a variant of the Aanderaa construction (see [1] or [26]) one can give a structural description of groups with word problem solvable in polynomial time. Let $G = \langle A \rangle$ be a finitely generated subgroup of a finitely presented group $H = \langle B \rangle$ and $A \subseteq B$. Then we define the *Dehn function $d_{G,H}(n)$ of G in H* as the smallest function $f(n)$ with the following property: for every number n and every word w over A which is equal to 1 in G, $|w| \leq n$, there exists a van Kampen diagram over H whose boundary label is w and area $\leq f(n)$; in other words, w is a product of at most $f(n)$ conjugates of the relators from H.

[1] Recall that we do not distinguish equivalent functions so $\mathcal{T}_4$ is actually the set of functions which are equivalent to time functions of Turing machines.

It is well known and easy to see ([9]) that the word problem of a finitely presented group is solvable if and only if the Dehn function is recursive. Nevertheless the difference between the computational complexity of the word problem and the Dehn function can be quite large (see [21]).

Notice that if $d(n)$ is the Dehn function of a finitely presented group G then there exists a "trivial" (non-deterministic) machine which checks if a word is equal to 1 in G. This machine simply inserts relators of G into the word until the word becomes 1. The time function of this machine is equivalent to $d(n)$ (for the precise description of the "trivial" machine and the proof that its time function is equivalent to $d(n)$ see in the proof of Theorem 1.1).

Thus the word problem in a group G can be decidable by a very fast machine but at the same time the "trivial" machine can be very slow. The following corollary shows that we can always embed G into another finitely presented group H such that the trivial machine for H solves the word problem in G almost as fast as the fastest Turing machine can.

Corollary 1.1 *The word problem in a finitely generated group G is solvable in polynomial time by a non-deterministic Turing machine if and only if G is embeddable into a finitely presented group H such that the Dehn function of G in H is bounded by a polynomial. Moreover, for every function $T(n) \geq n$, if the word problem in a group G is solvable in time $\leq T(n)$ by a non-deterministic Turing machine then G is embeddable into a finitely presented group H such that the Dehn function of G in H is bounded by $O(T(n)^4)$.*

One can also view this corollary as a result about distortions of areas of words in subgroups of finitely presented groups. By an *area* of a word $w = 1$ in G we mean the smallest area of a van Kampen diagram with boundary label w. Then the function $d_{G,H}(n)$ shows how distorted the areas of elements in the subgroup G of H are. Theorem 1.1 shows that the function $d_{G,H}(n)$ cannot be lower than the computational complexity $T(n)$ of the word problem for G. Corollary 1.1 shows that the distortion can reach as low as $T(n)^4$.

We are not going to prove Corollary 1.1 in this paper because in the next paper (joint with A. Yu. Ol'shanskii) we shall prove the following more powerful result.

Theorem 1.4 *The word problem of a group is decidable in polynomial time if and only if this group can be embedded into a group with polynomial Dehn function. Moreover every group with the word problem solvable in time $T(n)$ can be embedded into a group with isoperimetric function $T(n)^5$.*

The following corollary of Theorem 1.3 was the first result about Dehn functions of groups proved by Rips. It was the start point of this work.

Corollary 1.2 *(Rips) There exists a finitely presented group with undecidable conjugacy problem and Dehn function equivalent to n^3.*

The proof differs from the original proof of Rips and is given in the last section of this paper.

The next corollary shows how to use Theorems 1.2 and 1.3 in order to find many different Dehn functions and isodiametric functions.

With every function $f(n)$ one can associate two computational problems. It is not very easy to define computability of functions by non-deterministic machines. So when we talk about computability of functions, we restrict ourselves to deterministic (multi-tape) Turing machines (although one can replace them with non-deterministic machines if one gives a "right" definition of computability).

Problem A. Given a natural number n written in binary, compute $f(n)$ in binary. The size of n is the number of digits of n, that is $[\log_2 n] + 1$.

Problem B. Given a natural number n written in unary (as a sequence of 1's), compute $f(n)$ in unary. The size of n is n.

It is easy to see that if $f(n)$ is a function such that Problem B is solvable in time $O(f(n))$ then $f(n)$ is equivalent to the time function of a deterministic Turing machine (a Turing machine which computes $f(n)$).

Corollary 1.3 *Let $f(n) > n^4$ be a superadditive function such that the binary representation of $f(n)$ is computable in time $O(\sqrt[4]{f(n)})$ by a Turing machine (i.e., Problem A is solvable in time $O(\sqrt[4]{f(n)})$). Then $f(n)$ is equivalent to the Dehn function of a finitely presented group and the smallest isodiametric function of this group is equivalent to $T^{3/4}(n)$.*

Remark. Note that the size of n in Problem A is $O(\log n)$. Thus if, say, Problem A for a function $f(n)$ is solvable in polynomial time then the condition of the corollary holds and $f(n)$ is the Dehn function of some finitely presented group and the smallest isodiametric function of this group is equivalent to $T^{3/4}(n)$.

Proof. By Theorem 1.2 it is enough to show that $g(n) = [\sqrt[4]{f(n)}]$ is equivalent to the time function of a Turing machine.

Consider the following Turing machine M. The input of M is a natural number n written in unary. The machine has three tapes, with n (in unary) initially written on tape 1, and tapes 2 and 3 are empty. Machine M first computes the binary representation of n.

The algorithm is well known:

> Until the number n written on tape 1 is 0,
>   write the remainder of n modulo 2 on tape 2, $[n/2]$ in unary on tape 3,
>   erase tape 1,
>   copy $[n/2]$ from tape 3 to tape 1 and erase tape 3
> end of cycle.

This algorithm takes $O(n)$ steps. After this, M computes $f(n)$ in binary. By assumption, this takes at most $O(\sqrt[4]{f(n)}) = O(g(n))$ steps.

Then M computes $g(n) = [\sqrt[4]{f(n)}]$ in binary. Note that for every number m written in binary the number $[\sqrt[4]{m}]$ (in binary) can be computed in $O((\log m)^3)$ steps by the ordinary bisection algorithm. So M can compute the binary representation of $g(n)$ from the binary representation of $f(n)$ in $O((\log f(n))^3)$ steps.

Finally M computes $g(n)$ in unary. The algorithm is opposite to the one presented above. Again we use three tapes with $g(n)$ (in binary) initially written on tape 1, and with tapes 2 and 3 empty.

Write 1 on tape 2;
move the head one step right on tape 1;
until an empty square on tape 1 is to the right of the head, repeat:
    copy the content of tape 2 twice on tape 3 and erase tape 2
    copy the content of tape 3 on tape 2 and erase tape 3
    if 1 is written to the right of the head on tape 1 then write one more 1 on tape 2
    move the head one step right on tape 1
end of cycle

This takes at most $O(g(n))$ steps. Thus the whole algorithm is executed in time at most $O(n)+O(g(n))+O((\log f(n))^3)+O(g(n)) = O(g(n))$. This algorithm solves Problem B for $g(n)$. Therefore $g(n)$ is the time function of a Turing machine. By Theorem 1.2, $g(n)^4$ is equivalent to the Dehn function of a finitely presented group. Since $f(n)$ is equivalent to $g(n)^4$, we conclude that $f(n)$ is equivalent to the Dehn function of a finitely presented group. $\square$

This corollary allows one to construct finitely presented groups with "arbitrary weird" Dehn functions and smallest isodiametric functions. The following general fact follows from Corollary 1.3.

Corollary 1.4 *For every real number $\alpha \geq 4$ such that the first m digits of α can be computed in time $\preceq 2^{2^m}$ (for every m) the function $[n^\alpha]$ is equivalent to the Dehn function of a finitely presented group and the smallest isodiametric function of this group is $n^{3/4\alpha}$. On the other hand if n^α is the Dehn function of a finitely presented group then for every m the first m digits of α can be computed deterministically in time $\preceq 2^{2^{2^m}}$.*

Proof. Notice that the function $[n^\alpha]$ is equivalent to the function $2^{[\alpha[\log_2 n]]}$. The function $[\log_2 n]$ (i.e. the binary expression of the number of binary digits in n) is computable in time $\leq O((\log_2 n)^2)$ by an obvious algorithm: scan the number n from left to right on one tape and after each step add 1 to the number on the other tape.

Since the first $[\log_2(\log_2 n)] + 1$ digits of α are computable in time $O(n)$, the function $[\alpha[\log_2 n]]$ is computable in time $O(n) \leq O(n^{\alpha/4})$.

Notice also that Problem A for a function equivalent to 2^m is solvable in time $O(m)$. Indeed, we can consider the unary expression of m as the binary expression of $2^{m+1} - 1$ and use the second algorithm in the proof of Corollary 1.3. It remains to apply Corollary 1.3.

Now suppose that n^α is the Dehn function of a finitely presented group. Then by Theorem 1.1, n^α is equivalent to the time function $T(n)$ of some (non-deterministic)

Turing machine M. We can assume that when M accepts, all its tapes are empty and that M has only one accept configuration c_0. We can also assume that M has an input tape which has number 1.

By Gromov's theorem, we can assume that $\alpha \geq 2$. Then for any $n > 0$

$$\epsilon_1 n^\alpha \leq T(n) \leq \epsilon_2 n^\alpha$$

for some positive constants ϵ_1 and ϵ_2. Let number n_0 be such that $2^n > \log_2 \epsilon_2$ for every $n \geq n_0$. Let $q = [\alpha] + 1$.

Consider the following deterministic Turing machine M' which computes $T(n)$. Let k be the number of tapes of M. Then M' will have $k+3$ tapes. Tape $k+1$ will contain an input number n in binary, tape $k+2$ will contain $T(n)$ when M' stops, tape $k+3$ is auxiliary. In the input configuration, M' has number n written on tape $k+1$ and all other tapes empty. First of all it types the accept configuration of the machine M on the first k tapes and computes n^q (in binary) on tape $k+3$. Then it considers all possible computations of length $\leq n^q$ of the machine M^{-1} (commands of M are applied backwards) starting with the accept configuration c_0 of M. The number of such computations is at most r^{n^q} where r is the number of commands of M. For each of these computations C, it calculates the length $T(C)$ and the final word $W(C)$ written on the first tape. It writes this information on tape $k+2$ provided $|W(C)| \leq n$ and the final configuration of C is an input configuration of M. Thus it produces a sequence $\mathcal{S}$, consisting of numbers $T(C)$ and words $W(C)$. The words $W(C)$ in this sequence are all words of length $\leq n^q$ accepted by M. It is clear that the time to produce this sequence and the length of it does not exceed $Dn^q r^{n^q} \leq 2^{n^d}$ for some positive constants D, d, r. Then for every word W occurring in $\mathcal{S}$, M' calculates the minimal $t(W)$ of the corresponding numbers $T(C)$ (for all computations C such that $W = W(C)$). Then it finds the maximal number among all $t(W)$'s. This number is obviously equal to $T(n)$. After that M' writes $T(n)$ on tape $k+2$, erases all other tapes and stops. It is clear that in order to compute $T(n)$ given the sequence $\mathcal{S}$, M' needs time at most $|\mathcal{S}|^2 \leq 2^{n^{d_1}}$ for some constant d_1. Thus M' is a deterministic machine which computes $T(n)$ in time $\leq 2^{n^{d_2}}$ for some constant d_2.

Given this machine M', consider the following Turing machine M'' which will calculate the first m digits of α (for every m). This machine has $k+4$ tapes with tape $k+4$ being the input tape. It starts with number m in binary written on tape $k+4$ and all other tapes empty. Then it calculates the number $n = 2^{2^{m+n_0}}$ and writes it on tape $k+1$ (using tape $k+2$ as an auxiliary tape and cleaninig it after n is computed). Then M'' turns on the machine M' and produces $T(n)$ on tape $k+2$. Then it calculates $p = [(\log_2 T(n) - \log_2 \epsilon_1)/2^{n_0}]$ and writes it on tape $k+2$. Notice that $\alpha \log_2 n + \log_2 \epsilon_1 \leq \log_2 T(n) \leq \alpha \log_2 n + \log_2 \epsilon_2$. Therefore

$$[\alpha 2^m] \leq p \leq \alpha 2^m + (\log_2 \epsilon_2)/2^{n_0}.$$

Hence $p = [\alpha 2^m]$, so p is the number formed by the first $m+k$ binary digits of α where k is the number of digits in α before the period. From the construction of M'', it is clear that the time complexity of M'' does not exceed $2^{2^{d_2 2^{m+n_0}}} \leq 2^{2^{2^{d_3 m}}}$ for some constant d_3. $\square$

Notice that rational numbers ≥ 4 and fast computable irrational numbers like $\pi+1$, e^2, $e\pi$ and others satisfy the conditions of Corollary 1.4 (see [5]). It would be interesting to find out whether the number $2^{2^{2^m}}$ in the second statement of Corollary 1.4 can be decreased to 2^{2^m}. This would give a necessary and sufficient condition for n^α to be the Dehn function of a finitely presented group.

Corollary 1.4 implies that not every polynomially bounded increasing computable function $> n^2$ is the Dehn function of a finitely presented group. (just take a very slowly computable number α and consider the function n^α). This answers a question of Gersten.

The plan of the paper is the following. First we prove Theorem 1.1: every Dehn function of a finitely presented group is a time function of some Turing machine. This implies one inclusion in Theorem 1.2. In order to prove Theorem 1.3 and the other inclusion in Theorem 1.2 we start with an arbitrary multitape nondeterministic Turing machine with time function $T(n)$. Our goal is to simulate a Turing machine in a group. The mere fact that a Turing machine can be simulated by a group is not new. Papers by Novikov and Boone showed it long ago. The problem that we face in this paper is that we need the Dehn function of the group to be not much bigger than the time function of the Turing machine. In fact all known simulations of Turing machines in groups (see [17]) lead to groups with exponential Dehn functions. The same is true if we use Minsky machines instead of arbitrary Turing machines [17].

In order to overcome this difficulty, we introduce the concept of an S-machine. An S-machine is a kind of Turing machine with one tape and many heads (several heads can move at once). Heads are placed between cells on the tape, so we consider their states as letters when we talk about the word written on the tape. A typical command of an S-machine is $q \to apb^{-1}$ where q and p are states of some head and a is a tape symbol. Notice that this command does not depend on the content of the tape, only on the state of the head. It can be applied any time when the state of the head is q. Thus usually S-machines are extremely non-deterministic. The alphabet of S-machine is divided into subsets Y and Y^{-1} of the same size, so the words written on the tapes can be considered as group words. After every step of the machine, the word on the tape is automatically reduced. We do not consider reducing the word a separate step, it is a part of execution of a command.

For every Turing machine M with time function $T(n)$ we find an S-machine $\mathcal{S} = \mathcal{S}(M)$ which works in time $T(n)^3$ and recognizes almost the same language.

Then we define a finitely presented group $G_N(\mathcal{S})$ where N is a natural number. For $N = 1$ this presentation resembles the presentation of Boone's group (see [26]) but it does not have the Baumslag-Solitar type relations $xyx^{-1} = y^2$ which make the Dehn function exponential.

In fact the S-machines are needed precisely to avoid such relations.

We shall prove that (for N large enough), the Dehn function of $G_N(\mathcal{S})$ is $T(n)^4$, the smallest isodiametric function is $T^3(n)$ and that $G_N(\mathcal{S})$ satisfies the conditions of Theorem 1.3.

In order to do that, with every admissible word W of $\mathcal{S}$ we associate a word $K(W)$ over the generators of W. It is relatively easy to show that if W is accepted by $\mathcal{S}$ then there exists a "standard" van Kampen diagram (*disc*) with boundary label $K(W)$, so $K(W) = 1$

in $G_N(\mathcal{S})$. In order to compute the Dehn function and the smallest isodiametric function of $G_N(\mathcal{S})$ we consider an arbitrary van Kampen diagram Δ over the presentation of $G_N(\mathcal{S})$. We define a planar graph associated with this diagram, whose vertices are subdiscs of Δ. For N big enough the degree of each vertex of this graph is greater than 8, so small cancellation theory is applicable to this graph (see [20]). Using small cancellation theory, we decompose Δ in some natural way into simpler subdiagrams which either do not have subdiscs or are discs themselves. This is like decomposing a snowman into small snow balls. Then we show that the total area of the diagrams without discs is bounded by $|W|^4$ and the sum of the lengths of boundaries of discs is bounded roughly by $O(|W|)$. The area of a disc with perimeter n is bounded by the area of the corresponding computation of $\mathcal{S}$, that is by a function equivalent to $T(n)^4$ and its diameter is equivalent to $T^3(n)$. Using the fact that T^4 is superadditive, we conclude that the total area of the maximal discs inside Δ is bounded from above by a function equivalent to $T(n)^4$. This gives us the upper bound of the Dehn function.

To get the lower bound we consider an arbitrary van Kampen diagram with boundary label $K(W)$ where W is an admissible word for $\mathcal{S}$. We show that in this case W is accepted by $\mathcal{S}$ and that the area (diameter) of this diagram cannot be smaller than a constant times the area (diameter) of a disc corresponding to an accepting computation for W. This gives us the lower bound for the Dehn function and also proves that $K(W) = 1$ implies that W is accepted by $\mathcal{S}$.

In the last section of the paper we prove Corollary 1.2.

2 History

This paper has a long and complex history. Several versions of this paper containing gaps appeared as preprints during the last 2 years, and many people have these wrong versions. In this section, we try to explain what were the gaps and how these mistakes were fixed. Without it several main ideas of this paper would be unjustified. Another reason for writing this section was that different authors had different input in this paper.

We began working together on Dehn functions of groups during Rips' visit to Lincoln in Summer 1994. About a year before that Birget formulated the "embedding conjecture" that *every finitely presented group with word problem solvable in polynomial time can be embedded into a finitely presented group with polynomial Dehn function.* By that time the semigroup analog of this conjecture was proved by Birget. It appeared later in preprint [4]. In this preprint, Birget also proved semigroup versions of the main results of this paper about isoperimetric functions: analogs of Theorems 1.1, 1.2, 1.3. He also presented a sketch of a proof of our Theorem 1.1 for groups: every Dehn function of a finitely presented group is equivalent to the time function of some Turing machine (the complete proof presented here is due to Sapir).

Rips' idea of proving the "embedding conjecture" was the following. Consider the classic Novikov-Boone-Higman-Aanderaa embedding of a finitely generated group G into a finitely presented group H [26]. The goal was to show that the Dehn function of H is not much bigger than the time complexity of a machine M solving the word problem in G. This idea was explored before in Madlener and Otto [21] but the estimates of the

Dehn functions there were too rough, they dealt only with Dehn functions which are well above exponential. In fact the main problem is that the "classic" Novikov-Boone-Higman-Aanderaa construction produces groups with at least exponential Dehn function because of the relations of the Baumslag-Solitar form $a^{-1}xa = x^2$.

Recall that the Novikov-Boone-Higman-Aanderaa embedding has two steps [26]. In the first step one takes a Turing machine M and *simulates* it in a group $B(M)$. This means that there exists a map K from the set of configurations of M to the set of words in $B(M)$ such that $K(c) = 1$ in $B(M)$ if and only if the configuration c is accepted by M. In the second step one uses the Turing machine M solving the word problem of the group G and several relatively simple HNN-extensions of $B(M)$ to produce a finitely presented group H containing G.

Rips' program was, first, by modifying the Novikov-Boone construction get a group $B'(M)$ also simulating M with Dehn function polynomially related to the time function of M and, second, show that HNN-extensions of Aanderaa does not change the Dehn function too much.

In order to compute the Dehn function of a group Birget and Rips wanted to use geometric methods used by Rips in his proof of Corollary 1.2 and in his "geometrization" of the proof of Novikov-Boone theorem. This method counts the number of bands and annuli of cells in a van Kampen diagram, see Section 7 and Lemma 8.1 of this paper. Notice that a similar method was used in studying van Kampen diagrams over HNN-extensions by Miller and Schupp in 1973 [22].

In 1994, Birget and Rips decided that by removing the Baumslag-Solitar relations and modifying the Turing machine M instead, they can still simulate the machine M in a group $B'(M)$ without exponential blow up of the Dehn function. It also looked possible that the geometric methods will give not just an upper bound of the Dehn function of the group $B'(M)$ but the Dehn function up to the equivalence. This would give almost a complete description of all Dehn functions of finitely presented groups as time functions of Turing machines with some reasonable restrictions. This would solve a Short's problem of finding groups with exotic Dehn function, which Sapir was working on at that time. So Sapir joined Birget and Rips.

Trying to implement our program we constructed a modification $B'(M)$ of $B(M)$ and proved several geometric lemmas similar to those in Section 7 were proved.

Unfortunately these lemmas gave only an upper bound of the Dehn function which was way above the lower bound that we could get. It was not even clear that the Dehn function of $B'(M)$ is polynomial provided M has a polynomial time function. To compute the Dehn function up to equivalence, further modification of $B'(M)$ (with many sectors) and more geometric ideas (the snowman decomposition) were introduced and implemented mainly by Sapir. This gave us results similar to Theorems 1.2 and 1.3 of this paper only instead of exponent 4, we had exponent 3. The paper was written in February 1996 and Sapir put a link to it on his Web page. Many people downloaded the paper.

Immediately after that we started working on the second step of our program, the Higman-Aanderaa construction. Birget and Sapir introduced a many sectors modification of the Aanderaa construction and proved that this construction gives a Higman-type embedding of G into a finitely presented group H which was constructed as a sequence of HNN-extensions of $B'(M)$. But when we tried to prove that the Dehn function of H is not much bigger than the Dehn function of $B'(M)$, we discovered that one of the geometric lemmas (Lemma 5.11 in the 1996 version of our paper) proved among the first results in 1994 was wrong. In the "proof" of this lemma we used a property of the Turing machine M which we did not have. As a result not only the computation of the Dehn function of the group $B'(M)$ was wrong but even the fact that $B'(M)$ simulates M in the sense mentioned above proved to be incorrect.

At first this problem seemed easy to fix by modifying the Turing machine M. During seven months after that at least 20 different modifications of M were introduced. Each one of them seemed very promising at the beginning but turned out to be not better than the original modification later. The hope of ever fixing the mistake almost faded when Sapir realized that although the group $B'(M)$ does not simulate the Turing machine M, it still simulates a "machine" related to M. These machines were called S-machines. After that we faced a relatively standard problem in the theory of algorithms. We needed to prove that S-machines are polynomially equivalent to ordinary Turing machines. This turned out to be very non-trivial. Sapir's proof of this fact presented in this paper is probably the longest proof of equivalence of two computing devices in the theory of algorithms. After this proof was obtained, the geometric part of the paper, – both the snowman construction and the proof of the lower bound of the Dehn function, – needed modifications also. It was done by Sapir. He also proved Corollary 1.4 about Dehn functions of the form n^α. All results about isodiametric functions in this paper are formulated and proved by Sapir as well. Since the proofs of these results depend heavily on the technique used in dealing with Dehn functions, it was unreasonable to write a separate paper on isodiametric functions, so we included these results here.

Several comments by Ol'shanskii simplified the geometric part of our paper. We are grateful to him for these comments. We are also grateful to Victor Guba whose comments helped in writing the section about S-machines.

Ol'shanskii used several ideas of the 1996 version of our paper in his paper about distortions of subgroups in finitely presented groups [24]. He also used a many-sector variant of the Aanderaa embedding. During his visit to Lincoln in January 1997, he and Sapir realized that a combination of S-machines and geometric ideas of this paper and ideas of [24] gives a proof of Birget's "embedding conjecture". This completed the work started in 1994.

3 Turing Machines

In this section we collect all information about Turing machines that we need in the proof of our main results. We also prove Theorem 1.1.

We shall use the following standard notation for Turing machines. A (multi-tape) Turing machine has k tapes and k heads. One can view it as a six-tuple

$$M = \langle X, \Gamma, Q, \Theta, \vec{s}_1, \vec{s}_0 \rangle$$

where X is the input alphabet, Γ is the tape alphabet ($X \subseteq \Gamma$), $Q = \bigcup Q_i, i = 1, ..., k$ is the set of states of the heads of the machine, Θ is a set of transitions (commands), $\vec{s}_1$ is the k-vector of start states, $\vec{s}_0$ is the k-vector of accept states.

The number of tapes k of the machine is determined by Θ. We assume that in the normal situation the machine starts working with states of the heads forming the vector $\vec{s}_1$, with the head placed at the right end of each tape, and accepts if it reaches the state vector $\vec{s}_0$. In general, the machine can be turned on in any configuration and turned off at any time.

The leftmost square on every tape is always marked by α, the rightmost square is always marked by ω.

The head is placed between two consecutive squares on the tape. When we talk about the word written on the tape, we do not include α, ω, and the state of the head. We assume that the machine can insert or delete squares on the tape but only at the right end (just before the ω-sign).

At every moment the head observes two squares on each tape.

A *configuration* of a tape of a Turing machine is a word $\alpha uqv\omega$ where q is the current state of the head, u is the word to the left of the head and v is the word to the right of the head.

A *configuration* U of a Turing machine is a k-tuple

$$(U_1, U_2, ..., U_k)$$

where U_i is the configuration of tape i. The length $|U|$ of this configuration is the sum of lengths of the words U_i.

An *input configuration* is a configuration where the word written on the first tape is in X^+, all other tapes are empty, the head observes the right marker ω, and the states form the start vector $\vec{s}_1$. An *accept configuration* is any configuration where the state vector is $\vec{s}_0$, the accept vector of the machine.

A transition of a Turing machine is determined by the states of the heads and some of the $2k$ letters observed by the heads. As a result of a transition we can replace some of these $2k$ letters by other letters, insert new squares in some of the tapes and move the head one square to the left (right) with respect to some of the tapes.[2].

For example in a one-tape machine every transition is of the following form:

$$uqv \to u'q'v'$$

where u, v, u', v' are letters or empty words. The only constraints are that no letter can be inserted to the left of α or to the right of ω and the end markers cannot be deleted. This command means that if the state of the head is q, u is written to the left of q and v is written to the right of q then the machine must replace u by u', q by q' and v by v'.

For a general k-tape machine a command is a vector

$$\{U_1 \to V_1, ..., U_k \to V_k\}$$

where $U_i \to V_i$ is a command of a 1-tape machine, the elementary commands are listed in the order of tape numbers. In order to execute this command, the machine checks if U_i is a subword of the configuration of tape i ($i = 1, ..., k$), and then replaces U_i by V_i.

[2]In different books, one can find different definitions of Turing machines, but all these definitions are equivalent: the machines recognize the same languages, and have equivalent complexity functions.

Notice that for every command $\{U_1 \to V_1, ..., U_k \to V_k\}$, the vector $\{V_1 \to U_1, ..., V_k \to U_k\}$ is also a command of a Turing machine. These two commands are called *mutually inverse*.

A *computation* is a sequence of configurations $w_1, ..., w_n$ such that for every $i = 1, ..., n-1$ the machine passes from w_i to w_{i+1} by applying one of the transitions from Θ. A configuration w is said to be *accepted* by a machine M if there exists at least one computation which starts with w and ends with an accept configuration.

A word $u \in X^*$ is said to be *accepted* by the machine if the corresponding input configuration is accepted. The set of all accepted words over the alphabet X is called the *language accepted by the machine*.

Let $C = (c_1, ..., c_g)$ be a computation of a machine M such that for every $j = 1, ..., g-1$ the configuration c_{j+1} is obtained from c_j by a transition r_j from Θ. Then we call the word $r_1 \ldots r_{g-1}$ the *history* of this computation. The number g will be called the *time* (*duration*) of the computation. Let p_i ($i = 1, ..., g$) be the sum of the lengths of the configurations of the tapes in the configuration c_i. Then the sum of all p_i will be called the *area of computation* and will be denoted by $\mathrm{area}(C)$.

If the set of transitions Θ is divided into a number of disjoint parts $\bigcup_{i=1}^{n} \Theta_i$ then every computation C can be represented as a "concatenation" of computations

$$C_1 C_2 .... \tag{1}$$

where each C_i is a computation of one of the Θ_j, the last configuration of C_i is the first configuration of C_{i+1} and C_i are maximal blocks of C where rules from exactly one part C_j are applied. We shall use such representations often when a program of a Turing machine (and, later, a program of an S-machine) is divided into subroutines.

With every Turing machine one can associate five functions: the *time function* $T(n)$, the *space function* $S(n)$, the *generalized time function* $T'(n)$, the *generalized space function* $S'(n)$, and the *area function* $A(n)$. These functions will be called the *complexity functions* of the machine.

For every natural number n the number $T(n)$ is the minimal number p such that for every accepted input configuration w (with $|w| \leq n$) there exists at least one accepting computation of length at most p. The number $S(n)$ is the minimal natural number p such that for every accepted input configuration w (with $|w| \leq n$) there exists at least one accepting computation which contains only configurations of length $\leq p$. The definitions of $T'(n)$ and $S'(n)$ are similar but we consider arbitrary accepted configurations w of length n, not just the input configurations as in the definitions of $T(n)$ and $S(n)$. It is clear that $T(n) \leq T'(n)$ and $S(n) \leq S'(n)$ and it is easy to give examples where the inequalities are strict. The area function $A(n)$ is defined as the minimal number p such that for every accepted configuration w (with $|w| \leq n$) there exists at least one accepting computation of area at most p.

We do not only consider *deterministic* Turing machines, for example, we allow several transitions with the same left side. The following proof of Theorem 1.1 shows that nondeterministic machines arise naturally when one works with the word problem. In this proof we associate with every group G a nondeterministic Turing machine M such that M accepts those and only those words which are equal to 1 in G, and the time function of

M is equivalent to the Dehn function of G. Actually the machine simulates the standard way of deriving relations of a group from defining relations. The nondeterminism of the machine reflects the fact that for every word there are usually many relations applicable to this word. Computations of the machine M correspond to derivations in the group G. Although an idea of the proof of this statement can be found in Birget [4], here we present the first complete proof.

Let us recall the statement of Theorem 1.1. *The Dehn function of a finitely presented group is equivalent to the time function of a two-tape Turing machine. The language accepted by this machine coincides with the set of words equal to 1 in the group.*

Proof of Theorem 1.1. Let $G = \langle X \mid R\rangle$ be a finitely presented group. We assume that R is closed under taking inverses and cyclic shifts. Let U be the finite set of all pairs of words (u, v) from $(X \cup X^{-1})^*$ such that uv^{-1} is a relation from R. Consider a Turing machine M with alphabet $X \cup X^{-1}$, two tapes and the following program.

The program starts in state q_1 with the head at the right end on each tape, a word w written on the first tape and empty second tape. The head always stays next to the right marker during the work of our Turing machine. The alphabet is $X \cup X^{-1}$. The program consists of the following operations.

1 (move). If the word written on one of the tapes has letter a at the right end, the machine can erase the square containing this letter and insert a square with a^{-1} in it on the other tape. This operation requires 1 transition: $\{aq_1\omega \to q_1\omega, q_1\omega \to a^{-1}q_1\omega\}$

2 (substitution). If the word written on the first tape has suffix u and there exists a pair (u, v) from U then the machine can replace u by v, by deleting the $|u|$ squares on the first tape and inserting $|v|$ squares on the first tape containing the word v. It is easy to see that this operation requires at most $|u|+|v|$ transitions of the forms $\{aq_i\omega \to q_j\omega, q_i\omega \to q_j\omega\}$, $\{q_i\omega \to aq_j\omega, q_i\omega \to q_j\omega\}$.

3 (reduction). If the word written on the first tape and the word written on the second tape have the same rightmost tape letter, then the machine can erase the squares containing these letters. This operation requires one transition of the form $\{aq_1\omega \to q_1\omega, aq_1\omega \to q_1\omega\}$

4 (accept). The machine accepts if both tapes are empty.

It is easy to prove by induction that after any number of these operations the product of the word written on the first tape and the inverse of the word written on the second tape is equal to w in the group G. Therefore a word w is accepted by our machine only if it is equal to 1 in G.

Conversely, if a word w is equal to 1 in G then there exist a derivation $w \to w_1 \to \ldots \to 1$ such that at each step we either insert/delete a word $r \in R$ or insert/delete a subword of the form aa^{-1} where $a \in X \cup X^{-1}$. It is clear that there exists a computation of our Turing machine which simulates this derivation. Therefore w is accepted by the machine M. Thus the language accepted by M consists of all words which are equal to 1 in G.

Let us prove that the time function $t(n)$ of the machine M is equivalent to the Dehn function $d(n)$ of the group G. Let $(\alpha u_1p_1\omega, \alpha v_1p_1\omega), \ldots, (\alpha u_mp_m\omega, \alpha v_mp_m\omega)$ be an accepting computation of our machine where $u_1 = w$, $p_1 = q_1$, $v_1 = u_m = v_m = 1$, $p_m = q_0$, and for every $i = 1, \ldots, m-1$ the machine passes from the i-th configuration to the $i+1$-st configuration by executing one of the operations 1–4. Then there exists a derivation

$w \to w_1 \to \ldots \to 1$ of length at most m where at each step one of relations from R is applied. This implies that there exists a van Kampen diagram with at most m cells and boundary label w. Thus $t(n) \geq d(n)$ ($n = |w|$).

On the other hand suppose that there exists a van Kampen diagram Δ over the presentation of M with boundary label w. Using this diagram we shall construct an accepting computation of the machine M for the word w of length at most $d(|w|) + |w|$.

We shall call an edge of Δ a *bridge* if after removing this edge the graph Δ becomes a union of two connected components (it is clear that we won't get more than two connected components).

We shall describe a process of "deconstructing" Δ. For every $i = 1, 2, \ldots$, we define a diagram Δ_i, a path t_i consisting of bridges in Δ_i, a vertex v_i, a subdiagram Δ_i' of Δ_i, and two words x_i and y_i forming a configuration γ_i of the machine M. By induction on i we shall prove that for every $i \geq 1$:

a) v_i is the terminal vertex of t_i and initial vertex of Δ_i', the initial vertices of t_i and Δ_i coincide, t_i^{-1} is a subpath of $\partial(\Delta_i)$.

b) The path t_i has no common edge with Δ_i', t_i consists of bridges of Δ_i.

c) If $i > 1$ then the diagram Δ_i is a subdiagram of Δ_{i-1}.

d) The path t_i is simple.

e) If t_i is not empty then Δ_i' is a connected component of the graph obtained from Δ_i by removing the last edge of t_i.

f) If Δ_i has more than one vertex then either t_i is not empty or Δ_i' is not empty.

g) If $i > 1$ then γ_i is obtained from γ_{i-1} by one of the 3 operations of the machine M (move, substitution, reduction).

h) $y_i = \mathrm{Lab}(t_i)$, $x_i y_i^{-1}$ is the label of the contour of Δ_i.

Let $\Delta_1 = \Delta$, let v_1 be the initial vertex of Δ, let t_1 be an empty path with the initial vertex $\iota(t)$ and terminal vertex $\tau(t)$ equal to v_1. Let $\Delta_1' = \Delta_1$. Then $y_i = 1$ and $x_i = w$. This means that γ_1 is an initial configuration of the machine M. It is clear that properties a)–h) hold.

Assume that for some $i \geq 1$ we have defined Δ_i, Δ_i', t_i (v_i, x_i and y_i are determined by Δ_i, Δ_i' and t_i). If Δ_i consists of one vertex then the process stops. So suppose that Δ_i has more than one vertex.

1 (move). Suppose first that v is an initial vertex of a bridge e in Δ_i' such that $(t_i e)$ is a subpath of $\partial(\Delta_i)$. Then we set $v_{i+1} = \tau(e)$, $t_{i+1} = t_i e$. Since e is a bridge, removing this edge from Δ_i' divides Δ_i' into two connected components. The connected component containing $\tau(e)$ will be denoted by Δ_{i+1}'. We also let $\Delta_{i+1} = \Delta_i$. Notice that the properties a)–f) hold for the quadruple $(\Delta_{i+1}, \Delta_{i+1}', v_{i+1}, t_{i+1})$. The only non-obvious property is d), but it follows from the fact that $e \in \Delta_i$ and that t_i has no common edges with Δ_i (by the induction hypothesis). The configuration γ_{i+1} is obtained from γ_i by a move. Indeed,

notice that the subpath p of $\partial(\Delta)$ labelled by x_i is not empty (otherwise t_i would be a closed path, a contradiction with d)). Therefore e^{-1} is the last edge of p, so the label of e is the last letter of w. As a result of the operation, edge e^{-1} is subtracted from p and edge e is added to t_i. Thus (x_{i+1}, y_{i+1}) is obtained from (x_i, y_i) by removing the last letter of x_i and adding the inverse of this letter to y_i. This is the move operation. Thus properties g) and h) also hold.

2 (substitution). Now suppose that v_i is not an initial vertex of any bridge e in Δ_i' such that $(t_ie)^{-1}$ is a subpath of $\partial(\Delta_i)$ and suppose that there is an edge e in Δ_i' with initial vertex v_i and a cell π containing e. Let Δ_{i+1} (resp. Δ_{i+1}') be the result of removing e from Δ_i (resp. Δ_i'). We also let $v_{i+1} = v_i$, $t_{i+1} = t_i$. It is clear that properties a)–f) hold for the new triple $(\Delta_{i+1}, \Delta_{i+1}', t_{i+1})$. The path $\partial(\Delta_{i+1})$ is obtained from $\partial(\Delta_i)$ by replacing e with the rest of the boundary of the cell π. Thus x_{i+1} is obtained from x_i by replacing the last letter $\mathrm{Lab}(e)$ with a word r such that $\mathrm{Lab}(e)r^{-1}$ is a relation in R. Thus γ_{i+1} is obtained from γ_i by a substitution. Therefore properties g) and h) hold.

3 (reduction). Finally there exists a possibility that $\Delta_i' = \{v_i\}$. By e), since $\Delta_i \neq \{v_i\}$, t_i is not empty. Let e be the last edge of t_i. By e) v_i is a connected component of $\Delta_i \backslash \{e\}$. This means that the boundary of Δ_i has a subpath ee^{-1} or $e^{-1}e$. Then by removing the edge e and the vertex v_i from Δ_i we obtain a new diagram which will be denoted by Δ_{i+1}. Let v_{i+1} be the initial vertex of e and let t_{i+1} be the result of removing e from t_i. Now if t_{i+1} is empty then let $\Delta_{i+1}' = \Delta_{i+1}$. If t_{i+1} is not empty then let e' be the last edge of t_{i+1}. Since e' was a bridge in the diagram Δ_i, it is a bridge in the subdiagram Δ_{i+1}'. Therefore by removing e' from Δ_{i+1} we get a graph with two connected components. The connected component containing v_{i+1} is denoted by Δ_{i+1}'. Again it is clear that all properties a)–f) hold for the new triple $(\Delta_{i+1}, \Delta_{i+1}', t_{i+1})$. By assumption x_i and y_i have a common last letter $\mathrm{Lab}(e)$. The words x_{i+1} and y_{i+1} are obtained from x_i and y_i by removing this letter. This is a reduction. Therefore properties g) and h) hold.

Let E be the number of edges in Δ. Let us prove that by using at most $2E$ moves, substitutions and reductions, we can reduce Δ to a one-vertex diagram. Indeed, at every step we either increase the path t_i by one edge or delete at least one edge from the diagram. All these edges belong to the original diagram Δ. Since t_i is a simple path the number of edges of Δ which belong to at least one of the t_i's does not exceed E. The number of reductions and substitutions also cannot exceed E. Therefore after at most $2E$ steps the process must end. But the process ends only when Δ_i consists of one vertex. Therefore after at most $2E$ steps we reduce Δ to a one-vertex diagram.

Recall that the triple $(\Delta_1, \Delta_1', t_1)$ corresponds to the initial configuration γ_1 where $w = \mathrm{Lab}(\partial(\Delta))$ is written on the first tape and the second tape is empty. By property h) for every $i > 1$ the configuration γ_{i+1} is obtained from γ_i by one of the three operations (move, substitution, reduction). Since in the last triple Δ_i is a one vertex diagram, the last configuration has empty tapes, i.e. it is the accept configuration. The number of operations needed to transform γ_1 into the accept configuration is the same as the number of operations needed to "deconstruct" Δ, that is at most $2E$. Since each operation of M translates into a constant number of elementary transitions, there exists an accepting computation of M starting with γ_1 whose length does not exceed a constant times E. It is well known [20] that in any diagram Δ over a finite presentation the number of edges does not exceed the length of $\partial(\Delta)$ plus a constant multiple of the number of cells. Therefore the time function $t(n)$ of M does not exceed a constant times $n + d(n)$, where $d(n)$ is the Dehn function of G. It remains to notice that $d(n)$ is equivalent to $C(d(n) + n)$, for any constant C. The theorem is proved. $\Box$

Notice that the machine that we constructed in this proof has two tapes and it has no commands which move the head away from the endmarker ω. As we shall see later every Turing machine can be converted into a Turing machine recognizing the same language, having the same complexity functions, and having no head moving commands. On the other hand, it is easy to construct a one tape machine which is equivalent (in the above sense) to the machine constructed in this proof. One has to just concatenate the first tape of our machine with the symmetric image of the second tape (of course, one would also to remove the right endmarker of the first tape, the left endmarker on the second tape and identify the state letters on the two tapes). Thus *every Dehn function of a finitely presented group is equivalent to the time function of a one tape Turing machine.*

In the next section we will show how to convert any Turing machine to an S-machine The following lemma is a small preliminary step in this conversion.

Lemma 3.1 *For every Turing machine M recognizing a language L there exists a Turing machine M' with the following properties.*

1. *The language recognized by M' is L.*

2. *M' is symmetric, that is, with every command $U \to V$ it contains the* inverse *command $V \to U$.*

3. *The time, generalized time, space and generalized space functions of M' are equivalent to the time function of M. The area function of M' is equivalent to the square of the time function of M.*

4. *The machine accepts only when all tapes are empty.*

5. *Every command of M' or its inverse has one of the following forms for some i*

$$\{q_1\omega \to q_1'\omega, ..., q_{i-1}\omega \to q_{i-1}'\omega, aq_i\omega \to q_i'\omega, q_{i+1}\omega \to q_{i+1}'\omega, ...\} \tag{2}$$

$$\{q_1\omega \to q_1'\omega, ..., q_{i-1}\omega \to q_{i-1}'\omega, \alpha q_i\omega \to \alpha q_i'\omega, q_{i+1}\omega \to q_{i+1}'\omega, ...\} \tag{3}$$

   where a belongs to the tape alphabet of tape i, and q_j, q_j' are state letters of tape j. Thus if the head observes the right markers at the beginning of a computation, it will observe the right markers during the whole computation. If the head does not observe the right markers at the beginning, then no command is applicable, and so the computation is trivial.

6. *The letters used on different tapes are from disjoint alphabets. This includes the state letters.*

Proof. First we show how to construct a machine M' satisfying the first four properties. Then we shall show how to transform it into a machine satisfying all six properties.

The symmetrization of an arbitrary Turing machine is essentially well known (see [4]). We modify the construction from [4] here.

Let M be a k-tape Turing machine. We can assume without loss of generality that the first tape of M is the *input* tape, that is it can contain letters only from the input alphabet. If such a tape does not exist, we can always add it to the machine without changing the complexity functions or the language accepted by the machine.

Recall that in an *input configuration* of M, a word is written on the first tape, all other tapes are empty, the head is next to the right endmarker on each tape.

We can also assume that the machine M has only one *accept command*, that is a command in which the states in the right sides form the vector $\vec{s}_0$.

The machine M' has $k+1$ tapes and is a composition of three machines which we call *phases 1, 2 and 3*. The tape alphabet of the $k+1$-st tape is the set of commands Θ of the machine M. The tape alphabets of the other tapes are the same as in M. The input alphabet of M' coincides with the input alphabet of M.

In the first phase M' inserts a sequence of squares containing commands of machine M on tape $k+1$. The only constrain is that the first inserted square must contain the accept command.

Thus after the first phase, the machine M' has a sequence of commands of M written on the $k+1$-st tape. Then the machine checks if all tapes except tapes 1 and $k+1$ are empty, and proceeds to the second phase. This can be done by one transition of the form

$$\{q_1\omega \to q_1'\omega, \alpha q_2\omega \to \alpha q_2'\omega, ..., \alpha q_k\omega \to \alpha q_k'\omega, q\omega \to q'\omega\}. \tag{4}$$

In the second phase M' tries to execute on the first k tapes the sequence of commands written on tape $k+1$, reading them from the right to the left. In order to make the machine M' do this, we replace every command

$$\tau = \{U_1 \to V_1, ..., U_k \to V_k\}$$

of the machine M by the command

$$\tau' = \{U_1 \to V_1, ..., U_k \to V_k, \tau q \to q\tau\}$$

where q is the state of M' on tape $k+1$ (this state is not changed by any of the commands of M'). This command executes the transition τ on the first k tapes of M', checks if τ is written next to the left of the head on tape $k+1$ (if τ is not written there, the command τ' is not executed), and moves the head on tape $k+1$ one square to the left.

The second phase of the machine M' is deterministic. It is also *injective*, that is for every configuration of phase 2 there is at most one command of M' whose inverse is applicable to this configuration. Indeed, this command is determined by the letter on tape $k+1$ which is written next to q on the right.

If (and only if) it turns out that the sequence of commands written on tape $k+1$ during the first phase is a valid history of an accepting computation of machine M, that is if the head moves next to the left endmarker on tape $k+1$, the machine M' returns the head to the right endmarker on tape $k+1$ and passes to the third phase.

In the third phase the machine erases one by one all squares on all tapes and accepts. The third phase of M' is deterministic.

Let $\mathrm{Sym}(M')$ be the machine obtained by *symmetrizing* M', that is by adding the inverses to all commands of M'. Consider an arbitrary reduced computation of $\mathrm{Sym}(M')$. Since the second and the third phases of M' are deterministic and the first and the second phases are injective, every reduced computation of $\mathrm{Sym}(M')$ can be represented in the form (1):

$$C = C_{3,1}^{-1}C_{2,1}^{-1}C_{1,1}^{-1}C_{1,2}C_{2,2}C_{3,2}C_{3,3}^{-1}C_{2,3}^{-1}....$$

where $C_{i,j}$ is a computation of phase i. Some prefix and some suffix of this sequence may be empty, the other blocks are not empty. Now suppose that this computation is accepting. Then it contains a block $C_{3,j}^{\pm 1}$ for some j. Let ℓ be the length of the first configuration c in $C_{3,j}^{\pm 1}$. One can apply to the configuration c commands from phase 3 of M' and get to an accept configuration in fewer than ℓ steps because each step on phase 3 removes one square of the tapes of M'. Let $C'_{3,j}$ be the corresponding computation. It is clear that the suffix of the computation C after the configuration c has length not smaller than the length of $C'_{3,j}$. The area and the space of this suffix are also not smaller than the area and space of $C'_{3,j}$. So we can replace this suffix of the computation C by $C'_{3,j}$. As a result we shall get a computation with no greater length, area and space. Thus we can assume that C has only one block of the form $C_{3,j}^{\pm 1}$ and of course (since C is accepting) it should have the form $C_{3,j}$. This means that C has the form

$$C_{2,1}^{-1}C_{1,1}^{-1}C_{1,2}C_{2,2}C_{3,2}$$

where the first 3 blocks may be empty. Let ℓ be the length of the first configuration c in C.

If the blocks $C_{2,1}, C_{1,1}, C_{1,2}$ are empty then the length of the computation cannot exceed 2ℓ because the length of the block $C_{2,2}$ does not exceed the length of the word written on tape $k+1$ in c, and the length of $C_{3,2}$ does not exceed ℓ. The space of this computation does not exceed ℓ because the length of configurations does not get bigger during this computation. For the same reason the area does not exceed $O(\ell^2)$.

Suppose that $C_{1,2}$ is not empty and let c be the first configuration of $C_{1,2}$. Then we can apply inverses of the commands in phase 1 and get to the input configuration. Indeed, tapes $2, 3, ..., k$ in c must be empty, and the head must observe the right endmarker on each tape: otherwise the machine would not be able to execute the command (4) and pass from $C_{1,2}$ to $C_{2,2}$. Also since the machine passes from $C_{2,2}$ to the third phase, the first (from the left) square on tape $k+1$ must contain the accept command during the computation $C_{2,2}$. Since the content of tape $k+1$ does not change during the work of the second phase, the accept command must occupy the first square of tape $k+1$ at the end of the computation $C_{1,2}$. Therefore by applying inverses of the commands in phase 1, M' can erase all squares on tape $k+1$ and get to the input configuration of M'.

Let $(C'_{1,2})^{-1}$ be the corresponding computation. Then

$$C_{2,1}^{-1}C_{1,1}^{-1}(C'_{1,2})^{-1}C'_{1,2}C_{1,2}C_{2,2}C_{3,2}$$

is an accepting computation. The length and the space of the prefix $C_{2,1}^{-1}C_{1,1}^{-1}(C'_{1,2})^{-1}$ is $O(\ell)$ and the area of this prefix is $O(\ell^2)$. The first configuration c of $C'_{1,2}$ is an input configuration of M' whose length is $\ell' \leq \ell$. Since M' accepts this configuration, the sequence s of commands written on the $k+1$-st tape after $C_{1,2}$ is a history of accepting computation of the machine M. Therefore M accepts the word written on the first tape in c. Let m be the length of this history word s. From the description of M', it is clear that then the length of C is at most $O(\ell)+O(m)$, the space is at most $m+\ell$ and the area at most $O(m^2+\ell^2)$. Therefore if we replace s by the history word of a shortest computation of the machine M accepting the input word of configuration c, the length, space and area of the corresponding computation of M' cannot be greater then the length, space and area of C. Thus we can assume that $m = T(\ell')$, where T is the time function of the machine M. This implies that the length, space and area of C is, respectively, $O(T(\ell')+\ell)$, $O(T(\ell')+\ell)$, $O(T^2(\ell')+\ell^2)$. Since $\ell' \le \ell$, we can conclude that the time function, the generalized time function, the space function and the generalized function of M' are equivalent to the function T, and the area function of M' is equivalent to T^2.

We have also proved that every word accepted by $\mathrm{Sym}(M')$ is also accepted by M. The converse statement is obvious (in fact every word accepted by M is also accepted by M'). This proves that $\mathrm{Sym}(M')$ satisfies properties 1, 2, 3, 4 of the lemma.

Thus we can assume that the original machine M satisfies properties 1 – 4.

In order to get property 5, we divide every tape i of M into two tapes, and number them i and $i+1/2$. If a configuration of the old tape i is $\alpha uqv\omega$ then the configurations of the new tapes i and $i+1/2$ will be, respectively:

$$\alpha uq\omega$$

and

$$\alpha\bar{v}q\omega$$

where $\bar{v}$ is the word v written from right to left. Then we replace every command

$$\{u_1q_1v_1 \to u_1'q_1'v_1', ..., u_kq_kv_k \to u_k'q_k'v_k'\}$$

by the command

$$\{u_1q_1\omega \to u_1'q_1'\omega, \bar{v}_1q_1\omega \to \bar{v}_1'q_1'\omega, ..., u_kq_k\omega \to u_k'q_k'\omega, \bar{v}_kq_k\omega \to \bar{v}_k'q_k'\omega\}$$

Here $\bar{v}$ is either v, if v is not the right marker, or α if $v=\omega$. This vector of commands has $2k$ components listed in the order of the tape numbers (1, 3/2, 2, 5/2,...). It is clear that this machine recognizes the same language as M and has the same complexity functions.

Now one can replace each command

$$\{w_1q_1\omega \to w_1'q_1'\omega, ...\}$$

by a sequence of $2k$ commands of the form

$$\{q_1\omega \to q_1'\omega, ..., w_iq_i\omega \to w_i'q_i\omega, ...\} \tag{5}$$

(only the i-th component of this command has a non-state letter different from ω). This, of course, requires increasing the the set of states (each of these new commands uses new state letters). The application of each of these commands may change the word written on only one tape. Together these commands do what the initial command does.

This transformation does not change the language accepted by the machine, it does not decrease the complexity functions and can increase them not more than by a constant factor ($\leq 2k$). Thus we can assume that every command of M has the form (5).

Notice that if $w_i = \alpha$ then the command (5) is of the form (3) because w_i' is also equal to α. Otherwise we can replace this command by two commands

$$\{q_1\omega \to q_1''\omega, ..., w_iq_i\omega \to q_i''\omega, ...\}$$

and

$$\{q_1''\omega \to q_1'\omega, ..., q_i''\omega \to w_i'q_i'\omega, ...\}$$

where q_j'' are new state letters (which we add to the set of state letters). Notice that these commands have the desired form (2). It is clear that the new machine recognizes the same language and each complexity function can increase by a factor of at most 2. All of the commands of the new machine have the desired form (2) or (3)

The last property is easy to get. To get property 6, one just needs to replace the alphabet of each tape by a disjoint copy of this alphabet, and the set of state letters on each tape by a disjoint copy of this set. Then of course one needs to change the commands of the machine accordingly. $\Box$.

4 S-machines

In this section, we define S-machines and prove that S-machines are "polynomially" equivalent to multi-tape Turing machines.

We shall define S-machines as rewriting systems [17]. Let n be a natural number. A *hardware* of an S-machine is a pair (Y, Q) where Y is an n-vector of (not necessarily disjoint) sets Y_i, Q is a $(n+1)$-vector of disjoint sets Q_i, $\bigcup Q_i$ and $\bigcup Y_i$ are also disjoint. The elements of $\bigcup Y_i$ are called *tape letters*, the elements of $\bigcup Q_i$ are called *state letters*.

With every hardware $\mathcal{S} = (Y, Q)$ we associate the *language of admissible words* $L(\mathcal{S}) = Q_1F(Y_1)Q_2...F(Y_n)Q_{n+1}$ where $F(Y_j)$ is the language of reduced group words in the alphabet $Y_j \cup Y_j^{-1}$. This language completely determines the hardware. So instead of describing the hardware $\mathcal{S}$, one can describe the language of admissible words (which is in many cases more convenient).

If $0 \leq i \leq j \leq n$ and $W = q_1u_1q_2...u_nq_{n+1}$ is an admissible word then the subword $q_iu_i...q_j$ of W is called the (Q_i, Q_j)-subword of W $(i < j)$.

An S-machine with hardware $\mathcal{S}$ is a rewriting systems. The objects of this rewriting system are all admissible words.

The rewriting rules, or S-*rules*, have the following form:

$$[U_1 \to V_1, ..., U_m \to V_m]$$

where the following conditions hold:

Each U_i is a subword of an admissible word starting with a Q_ℓ-letter and ending with a Q_r-letter (where $\ell = \ell(i)$ must not exceed $r = r(i)$, of course).

If $i < j$ then $r(i) < \ell(j)$.

Each V_i is also a subword of an admissible word whose Q-letters belong to $Q_{\ell(i)}\cup\ldots\cup Q_{r(i)}$ and which contains a Q_ℓ-letter and a Q_r-letter.

If $\ell(1)=1$ then V_1 must start with a Q_1-letter and if $r(m)=n+1$ then V_n must end with a Q_{n+1}-letter (so tape letters are not inserted to the left of Q_1-letters and to the right of Q_{n+1}-letters).

To apply an S-rule to a word W means to replace simultaneously subwords U_i by subwords V_i, $i=1,...,m$. In particular, this means that our rule is not applicable if one of the U_i's is not a subword of W. The following convention is important:

After every application of a rewriting rule, the word is automatically reduced. We do not consider reducing of an admissible word a separate step of an S-machine.

For example, if a word is

$$q_1aaq_2bq_3ccq_4$$

and $q_i\in Q_i$, $a\in Y_1$, $b\in Y_2$, $c\in Y_3$ and the S-rule is

$$[q_1\to p_1a^{-1}, q_2bq_3\to a^{-1}p_2b'q_3c], \tag{6}$$

where $p_1\in Q_1$, $p_2\in Q_2$, $b'\in Y_2$, then the result of the application of this rule is

$$p_1p_2b'q_3cccq_4.$$

With every S-rule τ we associate the inverse S-rule τ^{-1} in the following way: if

$$\tau=[U_1\to x_1V_i'y_1,\ U_2\to x_2V_2'y_2,...,U_n\to x_nV_n'y_n]$$

where V_i' starts with a $Q_{\ell(i)}$-letter and ends with a $Q_{r(i)}$-letter, then

$$\tau^{-1}=[V_1'\to x_1^{-1}U_1y_1^{-1},\ V_2'\to x_2^{-1}U_2y_2^{-1},...,V_n'\to x_n^{-1}U_ny_n^{-1}].$$

For example, the inverse of the rule (6) is

$$[p_1\to q_1a, p_2b'q_3\to aq_2bq_3c^{-1}].$$

It is clear that τ^{-1} is an S-rule, $(\tau^{-1})^{-1}=\tau$, and that rules τ and τ^{-1} cancel each other (meaning that if we apply τ and then τ^{-1}, we return to the original word).

The following convention is also important:

We always assume that an S-machine is symmetric, that is if an S-machine contains a rewriting rule τ, it also contains the rule τ^{-1}.

As in the case of Turing machines, we can define the history of a computation of an S-machine as the sequence (word) of rules used in this computation. A computation is called *reduced* if the history of this computation is reduced, that is if two mutually inverse rules are never applied next to each other.

As usual the *length* of a computation $W_1,...,W_n$ is n, the *space* is $\max\{|W_i|\mid i=1,...,n\}$, and the *area* is $\sum_i|W_i|$.

We say that a computation of an S-machine is *proper* if no **new** negative letters appear in a word during this computation. We say that a computation is *semiproper* provided a new negative letter inserted during one step of the computation never disappears during the rest of the computation.

It is easy to see that if a computation

$$C = (W_1, W_2, ..., W_n)$$

is semiproper then the inverse computation

$$C^{-1} = (W_n, ..., W_2, W_1)$$

is also semiproper.

For example, let $k = 2$, $Y_1 = \{\delta\} = Y_2$, $Q_1 = \{\alpha\}$, $Q_2 = \{q_1, q_0\}$, $Q_3 = \{\omega\}$. Suppose further that the S-machine $\mathcal{S}$ has just four rules:

(1) $q_1 \to a^{-1} q_1 a$,

(2) $\alpha q_1 \to \alpha q_0$,

and their inverses.

Here is an example of a proper computation of this S-machine:

$$\alpha aaaq_1aa\omega \to \alpha aaq_1aaa\omega \to \alpha aq_1aaaa\omega \to \alpha q_1aaaaa\omega \to \alpha q_0aaaaa\omega.$$

The history of this computation is (1)(1)(1)(2).

Here is an example of a semiproper computation:

$$\alpha q_1\omega \to \alpha a^{-1}q_1a\omega \to \alpha a^{-1}a^{-1}q_1aa\omega.$$

The history of this computation is $(1)^{-1}(1)^{-1}$. It is also possible to prove that every reduced computation of this machine is semiproper.

There exists a "natural" way to convert a Turing machine M into an S-machine $\mathcal{S}$. The idea is simple. Take a Turing machine satisfying all conditions of Lemma 3.1. Concatenate all tapes of the machine M together and replace every command $aq\omega \to q'\omega$ by $q\omega \to a^{-1}q'\omega$, so that commands of M become S-rules. Unfortunately the S-machine $\mathcal{S}$ constructed this way does not inherit most of the properties of the original machine M.

Consider the following typical example. Let M be a one tape Turing machine with $X = Y = \{a\}$, $Q = \{q_1, q_2, q_0\}$ and the following commands:

1. $aq_1\omega \to q_2\omega$
2. $aq_2\omega \to q_2\omega$
3. $\alpha q_2\omega \to \alpha q_0\omega$

plus all inverses of these commands. It is obvious that this machine accepts the word a^n if and only if n is strictly positive. The corresponding S-machine has the following commands:

1. $q_1\omega \to a^{-1}q_2\omega$

2. $q_2\omega \to a^{-1}q_2\omega$

3. $\alpha q_2\omega \to \alpha q_2\omega$

and all inverse commands. This machine accepts the word $\alpha a^n q_1\omega$ for every integer n. For example, let $n=-1$. Then the computation is the following (we write the number of a command in parentheses; i^{-1} denotes the inverse of command number i):

$$\begin{array}{l}\alpha a^{-1}q_1\omega \to (1)\\ \alpha a^{-1}a^{-1}q_2\omega \to (2)^{-1}\\ \alpha a^{-1}q_2\omega \to (2^{-1})\\ \alpha q_2\omega \to (3)\\ \alpha q_0\omega\end{array}$$

In fact it is nontrivial to construct an S-machine which recognizes only positive powers of a. Such a machine will be the key block in the main construction of this section.

Let $\mathcal{S}$ be an S-machine, let W be and admissible word and let $w_1,...,w_k$ be some words. Then $\mathcal{S}(W,w_1,...,w_k)$ is the statement saying that there exists a computation of $\mathcal{S}$ starting with W and ending with a word containing $w_1,...,w_k$. The set of all such reduced computations will be denoted by $C\mathcal{S}(W,w_1,...,w_k)$. If this set has only one element, this element will also be denoted by $C\mathcal{S}(W,w_1,...,w_k)$. If C is a computation then τC is the last word of C.

We shall need the following two general results. They have a "diagram group nature" [14].

Lemma 4.1 *Let $\mathcal{S}$ be an S-machine, W and W' be admissible words and let W be positive. Suppose that there exists a computation of $\mathcal{S}$ connecting W and W'. Then if every reduced computation starting at W is semiproper then every reduced computation starting at W' is semiproper. In addition, suppose that every reduced computation C starting at W has length $\le f(|W|+|\tau C|)$ and area $\le g(|W|+|\tau C|)$ for some functions f and g. Then every reduced computation starting with W' has length $\le f(|W|+|\tau C|)+f(|W|+|W'|)$ and area $\le g(|W|+|\tau C|)+g(|W|+|W'|)$.*

Proof. Let C be a reduced computation connecting W and W' and let C' be a reduced computation starting with W'. Consider the concatenation of computations CC'. Let us represent C in the form C_1D and C' in the form $D^{-1}C_1'$ where the computation C_1C_1' is reduced. Then the computation C_1C_1' is semiproper since it starts at W.

Suppose that C' is not semiproper. Then a negative letter inserted during a step of C' is deleted during a later step of C'. These two steps cannot both occur in C_1' because then C_1C_1' would not be semiproper. Thus the insertion occurs in D^{-1}. But then D contains a step removing a negative letter. Since D is a suffix of the computation $C=C_1D$ and W is a positive word, this negative letter should have been inserted and removed during the computation C. Thus C is not semiproper, a contradiction.

The facts about the length and the area can be proved similarly. Indeed, the length (area) of C' does not exceed the sum of the lengths (areas) of C and the reduced form of the computation CC'. $\square$

Lemma 4.2 *Let $\mathcal{S}$ be an S-machine and W and W' be admissible words. Suppose that there exists a computation connecting W and W', W' contains subwords $w_1, ..., w_k$ and $C\mathcal{S}(W, w_1, ..., w_k)$ contains only one computation. Then $C\mathcal{S}(W', w_1, ..., w_k)$ contains only one (trivial) computation.*

Proof. Indeed, suppose that there exists a non-trivial reduced computation C' from $C\mathcal{S}(W', w_1, ..., w_k)$. Let C be the reduced computation connecting W and W'. Then the reduced form of the computation CC' is equal to C. Therefore the reduced form of C' is trivial, a contradiction (we assumed that C' is reduced and non-trivial). $\square$

We shall construct eleven S-machines $\mathcal{S}_1 - \mathcal{S}_9$, $\mathcal{S}_\alpha$, $\mathcal{S}_\omega$ one by one. Then we'll use these machines in order to construct an S-machine $\mathcal{S}(M)$ simulating a Turing machine M.

The machine $\mathcal{S}_1$ has the following hardware: $Y(1) = (\{\delta\}, \{\delta\}, \{\delta\}, \{\delta\})$, $Q(1) = (\{p_1, p_2, p_3\}, \{q_1, q_2, q_3\}, \{r_1, r_2, r_3\}, \{s_1, s_2, s_3\}, \{t_1, t_2, t_3\})$. The admissible words have the following form:

$$p\delta^{n_1}q\delta^{n_2}r\delta^{n_3}s\delta^{n_4}t$$

where p, q, r, s, t may have indices 1, 2 or 3.

The program $P(1)$ consists of the following rules and their inverses.

1. $[q_1 \to \delta^{-2}q_1\delta^2, r_1 \to \delta^{-1}r_1\delta]$

2. $[p_1q_1 \to p_2q_2, r_1 \to r_2, s_1 \to s_2, t_1 \to t_2]$

3. $[p_1\delta q_1 \to p_3\delta q_3, r_1 \to r_3, s_1 \to s_3, t_1 \to t_3]$

Lemma 4.3 *(Machine $\mathcal{S}_1$ divides a number by 2 and tells even from odd.) Let $n \geq 0$ and m be integers, $W = p_1\delta^n q_1 r_1 s_1 \delta^m t_1$ and $k = n/2$ if n is even and $k = (n-1)/2$ if n is odd. Let $W_2 = p_2q_2\delta^k r_2\delta^k s_2\delta^m t_2$, $W_3 = p_3\delta q_3\delta^k r_3\delta^k s_3\delta^m t_3$. Then*

(i) Every computation of $\mathcal{S}_1$ starting with W is semiproper.

(ii) Every word in any computation $W = U_1 \to U_2 \to ...$ has the form $p_i\delta^{\ell_1}q_i\delta^{\ell_2}r_i\delta^{\ell_2}s_i\delta^m t_i$ where $\ell_1 + 2\ell_2 = n$, $i = 1, 2, 3$.

(iii) $C\mathcal{S}_1(W, q_1r_1)$ consists of one (trivial) computation.

(iv) $\mathcal{S}_1(W, p_2)$ if and only if n is even. In this case $C\mathcal{S}_1(W, p_2)$ consists of one computation of length $n/2$ and $\tau C\mathcal{S}_1(W, p_2) = W_2$.

(v) $\mathcal{S}_1(W, p_3)$ if and only if n is odd. In this case $C\mathcal{S}_1(W, p_3)$ consists of one computation of length $(n-1)/2$ and $\tau C\mathcal{S}_1(W, p_3) = W_3$.

Proof. (i) Indeed, let $U_1 \to U_2 \to ...$ be a computation. Since rules 2 and 3 and their inverses do not insert (delete) δ, we can assume that in this computation, we apply only rule 1 and its inverse. Since our computation is reduced, an application of rule 1 (resp. its inverse) in our computation cannot be followed by an application of the inverse of rule 1 (resp. the rule 1 itself). Therefore, in this computation we apply either only rule 1 or only the inverse of rule 1. But rule 1 cannot remove a negative power of δ inserted by previous applications of rule 1 (because rule 1 always inserts δ^{-1} only to the left of q_1). Similarly the inverse of rule 1 cannot remove a negative power of δ inserted by a previous application of the inverse of rule 1. This proves that a negative letter which appears during our computation cannot disappear, so our computation is semiproper.

(ii) can be easily proved by induction: every rule of $\mathcal{S}_1$ preserves the property from (ii).

(iii) Suppose that a reduced computation C of $\mathcal{S}_1$ takes a word W to a word W' containing the subword q_1r_1. Then by (ii) W' must be equal to W. Since there is only one rule of $\mathcal{S}_1$ which can be applied to a word containing p_2 (resp. p_3) and our computation is reduced, we can assume that all of the words in the computation C contain p_1. Thus all rules applied in the computation are 1 or 1^{-1}. Since the computation is reduced, we can assume that only rule 1 is applied (then in the inverse computation only rule 1^{-1} is applied). But each application of rule 1 inserts a new δ between q_1 and r_1, so it is impossible that the last word in the computation contains the subword q_1r_1, a contradiction.

(iv), (v). The "if" parts of the first statement in (iv) and (v) are obvious. The "only if" part follows from (ii). Indeed (ii) shows that if $W = U_1 \to U_2, \ldots$ is a computation of $\mathcal{S}_1$ then U_j has the form $p_i\delta^{\ell_1}q_i\delta^{\ell_2}r_i\delta^{\ell_2}s_i\delta^m t_i$ where $\ell_1 + 2\ell_2 = n$, Now if n is odd then ℓ_1 stays always odd and no computation can take W to a word containing the subword p_1q_1. But if we never get this subword, we can never apply rule 2, so we never get a word containing p_2. Similarly if n is even, ℓ_1 stays always even, so we can never apply rule 3 and we can never get a word containing p_3.

Also from (ii), it follows that if C is a reduced computation of $\mathcal{S}_1$ starting with W and τC contains subword p_2q_2 then $\tau C \equiv W_2$; if τC contains subword p_3q_3 then $\tau C \equiv W_3$.

The fact that $C\mathcal{S}_1(W, p_2)$ and $C\mathcal{S}_1(W, p_3)$ contains at most one computation immediately follows from (iii). Indeed, suppose that, say, $C\mathcal{S}_1(W, p_2)$ contains two computations C_1 and C_2. By (iii) the end words in these computations must be the same. So we can consider the superposition $C_1C_2^{-1}$. This computation takes W to W, so by (iii) the reduced form of this computation must be trivial, so $C_1 = C_2$. $\square$

The hardware of the machine $\mathcal{S}_2$ is the following:

$$Y(2) = Y(1), Q(2) = (\{p_1, p_2\}, \{q_1, q_2\}, \{r_1, r_2\}, \{s_1, s_2\}, \{t_1, t_2\}).$$

The program $P(2)$ consists of the following rules and their inverses:

1. $[q_2 \to \delta q_2\delta^{-1}, s_2 \to \delta^{-1}s_2\delta]$

2. $[q_2r_2s_2 \to q_1r_1s_1, p_2 \to p_1, t_2 \to t_1]$

The following lemma can be proved similarly to Lemma 4.3, so we omit the proof.

Lemma 4.4 *(Machine $\mathcal{S}_2$ moves q and s toward each other until they meet at r.) Let $k, \ell, m \geq 0$, n be integers, $W = p_2\delta^k q_2\delta^\ell r_2\delta^m s_2\delta^n t_2$. Let $W_1 = p_1\delta^{k+\ell}q_1r_1s_1\delta^{m+n}t_1$. Then*

(i) Every reduced computation of $\mathcal{S}_2$ starting with W is semiproper.

(ii) $C\mathcal{S}_2(W, W)$ consists of one (trivial) computation.

(iii) $\mathcal{S}_2(W, p_1)$ *if and only if* $k = \ell$. *In this case* $C\mathcal{S}_2(W, p_1)$ *consists of one computation and* $\tau C\mathcal{S}_2(W, p_1) = W_1$.

The machine $\mathcal{S}_3$ is a *cycle* of machines $\mathcal{S}_1$ and $\mathcal{S}_2$. One can describe $\mathcal{S}_3$ as the machine obtained by taking the union of $\mathcal{S}_1$ and $\mathcal{S}_3$ and identifying two state vectors of $\mathcal{S}_1$ with two state vectors of $\mathcal{S}_2$. This machine will divide a number (the exponent of δ) until it gets an odd number. In case when this exponent is 0, it will never stop. Thus $\mathcal{S}_3$ will tell non-zero numbers from zero.

More precisely, the hardware $(Y(3), Q(3))$ of the machine $\mathcal{S}_3$ is the same as the hardware of $\mathcal{S}_1$. The program $P(3)$ consists of the following rules and their inverses. We divide the program into two subprograms which we call *steps*:

Step 1.

1.1 $[q_1 \to \delta^{-2} q_1 \delta^2, r_1 \to \delta^{-1} r_1 \delta]$

1.2 $[p_1 q_1 \to p_2 q_2, r_1 \to r_2, s_1 \to s_2, t_1 \to t_2]$

1.3 $[p_1 \delta q_1 \to p_3 \delta q_3, r_1 \to r_3, s_1 \to s_3, t_1 \to t_3]$

Step 2.

2.1 $[q_2 \to \delta q_2 \delta^{-1}, s_2 \to \delta^{-1} s_2 \delta]$

2.2 $[q_2 r_2 s_2 \to q_1 r_1 s_1, p_2 \to p_1, t_2 \to t_1]$

Notice that Step 1 has the same rules as the machine $\mathcal{S}_1$ and Step 2 has the same rules as the machine $\mathcal{S}_2$.

Lemma 4.5 *(The machine $\mathcal{S}_3$ tells zero from non-zero.) Let $n \geq 0$ and either $n = 0$ or $n = 2^e(2m+1)$. Let $W = p_1 \delta^n q_1 r_1 s_1 t_1$. Let $W_3 = p_3 \delta q_3 \delta^m r_3 \delta^m s_3 \delta^{n-2m-1} t_3$. Then*

1. *Every reduced computation C of $\mathcal{S}_3$ starting with the word W or W_3 is semiproper. The length of C does not exceed $O(|W|+|V|)$ and the area does not exceed $O((|W|+|V|)^2)$ where $V = \tau C$.*[3]

2. *Each of the sets $C\mathcal{S}_3(W, s_1 t_1)$ and $C\mathcal{S}_3(W_3, p_3)$ consists of one (trivial) computations*

3. *$\mathcal{S}_3(W, p_3)$ if and only if $n > 0$. In this case $C\mathcal{S}_3(W, p_3)$ consists of one computation and $\tau C\mathcal{S}_3(W, p_3) = W_3$. The length of $C\mathcal{S}_3(W, p_3)$ is $O(n)$ and the area is $O(n^2)$.*

Proof. In order to prove the lemma, we shall describe all possible computations starting with the word W. Notice that every computation C of our machine has a representation in one of the following forms

$$C_{1,1} C_{2,1} C_{1,2} C_{2,2} \ldots \tag{7}$$

[3]In this paper we write that $f(n) = O(g(n))$ if $C_1 g(n) \leq f(n) \leq C_2 g(n)$ for some positive constants C_1, C_2 and every n. We write $f(n) \leq O(g(n))$ if $f(n) \leq C g(n)$ for some constant C and every n.

or

$$C_{2,1}C_{1,2}C_{2,2}\dots. \tag{8}$$

where $C_{i,j}$ is a computation of step i, $i=1,2$. Without loss of generality we can assume that each $C_{i,j}$ is non-trivial. The last word in each of these blocks (except the last one) is the first word of the next block.

If C consists of one block then all statements of our lemma follow from Lemmas 4.3 and 4.4. So suppose that C has more than 1 block.

Case 1. Suppose that $n>0$ and C has the form (7). If the last word U in $C_{1,1}$ contains p_1 then the first rule applied in $C_{2,1}$ is rule 2.2^{-1}. This means that U contains the subword $q_1r_1s_1$. By Lemma 4.3 (iii) in this case $C_{1,1}$ is trivial, a contradiction. Therefore U contains p_2. By Lemma 4.3(iv), U has the form

$$p_2q_2\delta^{n/2}r_2\delta^{n/2}s_2t_2$$

and n is even. Since the computation $C_{1,1}$ is semiproper (Lemma 4.3(i)) and U is a positive word, this computation is proper. By Lemma 4.3(iv) there is only one computation in $C\mathcal{S}_1(W,p_2)$. It is clear that the history of this computation is $(1.1)^{n/2}(1.2)$ (meaning that first we apply rule (1.1) $n/2$ times and then rule (1.2)).

Similarly, if a part $C_{1,i}$ of C starts with a word

$$p_1\delta^kq_1r_1s_1\delta^\ell t_1,$$

$k\geq 0,\ell\geq 0$ and is followed by $C_{2,i}$ then k is even, the computation $C_{1,i}$ is proper, the last word in $C_{2,i}$ is

$$p_2q_2\delta^{k/2}r_2\delta^{k/2}s_2\delta^\ell t_2$$

and the history of $C_{1,i}$ is $(1.1)^{k/2}(1.2)$.

Now suppose that a part $C_{2,i}$ starts with a word of the form

$$U=p_2q_2\delta^kr_2\delta^ks_2\delta^\ell t_2$$

and there exists $C_{1,i+1}$. Let V be the last word in $C_{2,i}$.

Suppose that V contains p_2. Then the first rule in $C_{1,i+1}$ is rule 1.2^{-1}. Therefore V contains p_2q_2. Thus V has the form

$$p_2q_2\delta^{k_1}r_2\delta^{k_2}s_2\delta^{\ell'}t_2.$$

Since r_2 and p_2 do not move during the computation of step 2, $k_1+k_2=2k$. Since t_2 also does not move, we have: $\ell'=\ell$. Finally, it is easy to prove by induction that during a computation of step 2 the difference between the distance from q_2 to r_2 and the distance from r_2 to s_2 stays the same. Since this difference is 0 in the word U, we have that $k_1=k_2$. This implies that $k_1=k_2=k$, $\ell'=\ell$, so $U=V$. By Lemma 4.4(ii) $C_{2,i}$ is trivial, a contradiction.

Thus V contains p_1. By Lemma 4.4(iii) V must have the form

$$p_1\delta^kq_1r_1s_1\delta^{k+\ell}t_1.$$

Since this word is positive and the computation is semiproper (Lemma 4.4(i)) the computation is proper. By Lemma 4.5(iii) there exists only one computation which takes U to V. It is clear that the history of this computation is $(2.1)^k(2.2)$.

Let N be the number of blocks in the representation (7) of C. Combining the information obtained so far, we can conclude that the history of the first $N-1$ blocks has the form:

$$(1.1)^{n/2}(1.2)(2.1)^{n/2}(2.2)(1.1)^{n/4}(1.2)....$$

Now suppose that the last block C' in (7) is a computation of step 1. By what we have proved before, C' starts with the word

$$p_1\delta^k q_1 r_1 s_1 \delta^{n-k} t_1$$

where $k = n/(2^c)$, $c = (N-1)/2$ (in this case N is odd). By Lemma 4.3(iv) if C' ends with a word containing p_2 then k is even and the history of C' is $(1.1)^{k/2}(1.2)$. If C' ends with a word containing p_3 then k is odd, that is $k = 2m+1$, $c = e$ (recall that $n = 2^e(2m+1)$) and the history of C' is $(1.1)^m(1.3)$. If this computation ends with a word containing p_1 then the computation C' is semiproper and the history is $(1.1)^d$ for some (not necessarily positive) integer d.

Thus we proved that if $V = \tau C$, C has the form (7), $n > 0$, and the last block in C is a computation of step 1, then $C\mathcal{S}_3(U,V)$ contains only one reduced computation of the form (7) which is semiproper if V contains p_1 and is proper if V contains p_2 or p_3. The word V contains p_3 if and only if $V = W_3$.

Suppose now that C' is a computation of step 2. Then it starts with the word

$$p_2 q_2 \delta^k r_2 \delta^k s_2 \delta^{n-2k} t_2$$

where $k = n/(2^c)$, $c = N/2$ (in this case N is even). If C' ends with a word containing p_1 then it's history has the form $(2.1)^k(2.2)$. Otherwise the history of C' has the form $(2.1)^d$ for some integer d.

Let again $V = \tau C$. Then if C has the form (7), $n > 0$, and the last block of C is a computation of step 2, then V can contain p_1 or p_2 (it cannot contain p_3), $C\mathcal{S}_3(U,V)$ contains only one reduced computation of the form (7) and C is semiproper if V contains p_1, and it is proper if V contains p_2.

We have also proved that in this case the computation C has one of the following histories:

$$(1.1)^{n/2}(1.2)(2.1)^{n/2}(2.2)(1.1)^{n/4}(1.2)...(1.1)^{k/2}(1.2) \tag{9}$$

where $k = n/2^c, c = (N-1)/2$;

$$(1.1)^{n/2}(1.2)(2.1)^{n/2}(2.2)(1.1)^{n/4}(1.2)...(1.1)^m(1.3) \tag{10}$$

where the number of blocks is $2e+1$;

$$(1.1)^{n/2}(1.2)(2.1)^{n/2}(2.2)(1.1)^{n/4}(1.2)...(2.2)(1.1)^d \tag{11}$$

where the number of blocks is $\leq 2e+1$;

$$(1.1)^{n/2}(1.2)(2.1)^{n/2}(2.2)(1.1)^{n/4}(1.2)...(1.2)(2.1)^{k}(2.2) \tag{12}$$

where the number of the blocks is $\leq 2e$;

$$(1.1)^{n/2}(1.2)(2.1)^{n/2}(2.2)(1.1)^{n/4}(1.2)...(1.2)(2.1)^{d} \tag{13}$$

where the number of the blocks is $\leq 2e$.

It is easy to check that each of these computations has length $O(|W|+|V|)$ and area $O((|W|+|V|)^2)$ where V is the last word in the computation.

Case 2. Suppose that $n>0$ and C has the form (8). Then the first rule applied in $C_{2,1}$ is $(2.2)^{-1}$ because the other command of step 2 is not applicable.

Therefore the second word in $C_{2,1}$ is

$$U_1 = p_2\delta^n q_2 r_2 s_2 t_2.$$

If the block $C_{2,1}$ ends with a word containing p_1 then by Lemma 4.4 this word must have the form

$$p_1\delta^n q_1 r_1 s_1 t_1$$

and the computation $C_{2,1}$ is proper. This means that rules (2.1), $(2.1)^{-1}$ are not applied in $C_{2,1}$, so the second rule applied in $C_{2,1}$ is rule 2.2, and this computation is not reduced, a contradiction.

Thus the last word in $C_{2,1}$ must contain p_2. Then the first rule applied in $C_{1,2}$ should be the rule $(1.2)^{-1}$. Thus the last word in $C_{2,1}$ must contain p_2q_2. By Lemma 4.4 the computation C must therefore start with an application of rule $(2.2)^{-1}$ and after that rule $(2.1)^{-1}$ is applied n times (each application of rule $(2.1)^{-1}$ moves q_2 one step closer to p_2). Hence the last word in $C_{2,1}$ and the first word in $C_{1,2}$ is

$$V_1 = p_2q_2\delta^n r_2\delta^n s_2\delta^{-n}t_2.$$

Now the first rule applied in $C_{1,2}$ must be $(1.2)^{-1}$. So the second word in $C_{1,2}$ must be

$$p_1q_1\delta^n r_2\delta^n s_2\delta^{-n}t_2.$$

By Lemma 4.3 the computation $C_{1,2}$ is semiproper, so in the rest of $C_{1,2}$ only one rule applies (several times): (1.1) or $(1.1)^{-1}$. If the rest of $C_{1,2}$ consists of applications of rule (1.1) then the rest of $C_{1,2}$ cannot have a word containing either p_1q_1 or $q_1r_1s_1$. Therefore C does not have block $C_{2,2}$.

Thus if $C_{2.2}$ exists then $C_{2,2}$ must start with

$$p_1\delta^{2n}q_1r_1s_1\delta^{-n}t_1$$

and the history of $C_{1.2}$ has the form $(1.2)^{-1}(1.1)^{-n}$.

Continuing in this manner (induction on the length of C) one can prove that the history of computation C has one of the following forms:

$$(2.2)^{-1}(2.1)^{-n}(1.2)^{-1}(1.1)^{-n}(2.2)^{-1}(2.1)^{-2n}(1.2)^{-1}(1.1)^{-2n}...(2.2)^{-1}(2.1)^{k} \tag{14}$$

or

$$(2.2)^{-1}(2.1)^{-n}(1.2)^{-1}(1.1)^{-n}(2.2)^{-1}(2.1)^{-2n}(1.2)^{-1}(1.1)^{-2n}...(1.2)^{-1}(1.1)^{k} \qquad (15)$$

where k is an integer.

This immediately implies that the length of C is $O(|W|+|V|)$ and the area of C is $O((|W|+|V|)^2)$ where $V=\tau C$.

Also if $V=\tau C$, $n>0$, and C has the form (8) then V cannot contain p_3 and $C\mathcal{S}_3(U,V)$ consists of one semiproper computation. It is clear that this V cannot be reached from W by any computation of the form (7). It is also clear that any word reachable by a computation of the form (8) cannot be reached by a computation of the form (7).

Comparing our results in cases 1 and 2, we can conclude that if $V=\tau C$, $n>0$ then $C\mathcal{S}_3(W,V)$ contains only one reduced computation, this computation is semiproper if V contains p_1 or p_2, and it is proper if V contains p_3. Moreover there exists only one reduced computation in $C\mathcal{S}_3(W,p_3)$. This computation has the form (7) and the last word in this computation is W_3.

Case 3. Finally consider the case when $n=0$. Then $W=p_1q_1r_1s_1t_1$. Then it is easy to see using Lemmas 4.3 and 4.4 that every block $C_{i,j}$ in the computation C, except the last one, has only two words, so the history of each block is of length one and the command used in each of the blocks except for the last one is either $(1.2)^{\pm 1}$ or $(2.2)^{\pm 1}$. Only the indices of letters change during the computation C until the last block. Thus the last block starts either with $p_1q_1r_1s_1t_1$ or with $p_2q_2r_2s_2t_2$. In the first case the last block either has two words and the rule applied in the last block is 1.2, 2.2^{-1}, or it contains more than 2 words and the rule applied in the last block is $1.1^{\pm 1}$. In the second case, either the last block has 2 words and the rule is 2.2 or 1.2^{-1}, or it contains more than two words and the rule applied in this block is $2.1^{\pm 1}$. Thus the computation C has one of the following forms:

$$((1.2)(2.2))^k \qquad (16)$$

for some integer k;

$$((1.2)(2.2))^k(1.1)^m \qquad (17)$$

for some integers k and m;

$$((1.2)(2.2))^k(1.2)^{\pm 1} \qquad (18)$$

for some integer k;

$$((1.2)(2.2))^k(1.2)^{\pm 1} \qquad (19)$$

for some integer k.

Since k and m are $O(|V|)$, in each case the length of the computation is $O(|W|+|V|)$ and the area is $O((|W|+|V|)^2)$.

This completes the description of C.

Thus if $n=0$ then the set of computations $C\mathcal{S}_3(W,p_3)$ is empty, and every computation starting with W is semiproper.

Now all statements of the lemma can be proved easily.

(i) We have proved that every computation starting with the word W is semiproper and the length and area of it satisfies condition 1 of the lemma. The fact that every computation starting with W_3 satisfies the same properties follows from Lemma 4.1.

(ii) The first part follows from the description of all computations starting at W. The second part follows from Lemma 4.2.

(iii) This has been established above. The length of $C\mathcal{S}_3(W, p_3)$ is $O(n/2+n/4+...) = O(n)$ and the area is $O(n^2/2 + n^2/4 + ...) = O(n^2)$. $\square$

The machine $\mathcal{S}_4$ is a *concatenation* of two copies of $\mathcal{S}_3$ with common states p_3, q_3, r_3, s_3 and t_3. More precisely, take a copy $\mathcal{S}_3'$ of the machine $\mathcal{S}_3$. The machine $\mathcal{S}_3'$ is obtained by adding $'$ to all state letters of $Q(3)$ except p_3, q_3, r_3, s_3 and t_3. Thus $\mathcal{S}_3'$ has states $Q'(3) = \{p_1', p_2', p_3\} \cup \{q_1', q_2', q_3\} \cup \{r_1', r_2', r_3\} \cup \{s_1', s_2', s_3\} \cup \{t_1', t_2', t_3\}$.

Now take the union $Q(4)$ of $Q(3)$ and $Q'(3)$ and consider the union $P(4)$ of programs $P(3)$ and $P'(3)$ of the machines $\mathcal{S}_3$ and $\mathcal{S}_3'$. The new machine with the hardware $(Y(1), Q(4))$ and program $P(4)$ will be denoted by $\mathcal{S}_4$.

Lemma 4.6 *(The machine $\mathcal{S}_4$ tells zero from non-zero and returns all state letters to their original positions.) Let $W = p_1\delta^n q_1 r_1 s_1 t_1$, $n \geq 0$, $W' = p_1'\delta^n q_1' r_1' s_1' t_1'$.*

(i) Every reduced computation C of $\mathcal{S}_4$ starting with W or W' is semiproper. The length of C does not exceed $O(|W| + |V|)$ and the area does not exceed $O((|W| + |V|)^2)$ where $V = \tau C$.

(ii) Each of the sets $C\mathcal{S}_4(W, s_1t_1)$, $C\mathcal{S}_4(W', s_1't_1')$ consists of one (trivial) computation.

(iii) $\mathcal{S}_4(W, s_1't_1')$ if and only if $n > 0$. In this case $C\mathcal{S}_4(W, s_1't_1')$ consists of one computation and $\tau C\mathcal{S}_4(W, s_1't_1') = W'$. The length of this computation is $O(n)$.

(iv) $\mathcal{S}_4(W', s_1t_1)$ if and only if $n > 0$. In this case $C\mathcal{S}_4(W', s_1t_1)$ consists of one computation and $\tau C\mathcal{S}_4(W', s_1t_1) = W$. The length of this computation is $O(n)$ and the area is $O(n^2)$.

Proof. All statements of the lemma will again follow from a description of computations of $\mathcal{S}_4$ starting with W.

Every computation C of $\mathcal{S}_4$ can be represented in the form $C_1C_2C_3\ldots$ where C_i is a computation of one of the machines $\mathcal{S}_3$ or $\mathcal{S}_3'$, neighbor blocks are computations of different machines, and each block is non-trivial.

Since C starts with the word W, C_1 must be a computation of the machine $\mathcal{S}_3$. If $C = C_1$ then C is a computation of $\mathcal{S}_3$ and we can apply Lemma 4.5. It is clear that in this case τC cannot contain p_1'.

If $C \neq C_1$ then C_2 is a computation of $\mathcal{S}_3'$. Therefore C_2 starts with a word U containing p_3. This word is the end word of C_1. So statement $\mathcal{S}_3(W, p_3)$ holds and by Lemma 4.5(iii) $n > 0$, $U = p_3\delta q_3\delta^m r_3\delta^m s_3\delta^{n-2m-1}t_3$ and there exists a unique reduced computation of $\mathcal{S}_3'$ which takes U to W'. Since U is a positive word, the computation C_1 is proper (Lemma 4.5(i)). By Lemma 4.5(i) the computation C_2 is semiproper.

Suppose that C_3 exists. Then it must be a computation of $\mathcal{S}_3$. Therefore it must start with a word containing p_3. By Lemma 4.5(ii) then C_2 is trivial, a contradiction.

Thus C consists of one or two blocks: $C = C_1$ or $C = C_1C_2$. In both cases C is semiproper. The computation C can contain two blocks if and only if $n > 0$.

Now all statements of the lemma follow immediately from this description and Lemmas 4.1 and 4.2. $\Box$

We shall need a copy of $\mathcal{S}_4$, $\bar{\mathcal{S}}_4$ obtained from $\mathcal{S}_4$ by putting $^-$ over all its state letters.

Now take an arbitrary alphabet Y and consider the following machine $\mathcal{S}_5$. The admissible words for $\mathcal{S}_5$ have the following form:

$$EuxvFE'p\Delta_1q\Delta_2r\Delta_3s\Delta_4t\bar{p}\Delta_5\bar{q}\Delta_6\bar{r}\Delta_7\bar{s}\Delta_8\bar{t}F'$$

where $E, x, F, E', p, q, r, s, t, \bar{p}, \bar{q}, \bar{r}, \bar{s}, \bar{t}, F'$ are state letters from different Q-components, $u \in (Y \cup Y^{-1})^*$, $\Delta_j \in \langle\delta\rangle$. If z is one of the letters $p, q, r, s, t, \bar{p}, \bar{q}, \bar{r}, \bar{s}, \bar{t}$ then z belongs to a Q-component $\{z_0, z_1, z_2, z_3, z_4, z_0', z_1', z_2'\}$. The alphabet of tape letters also includes letter x_4 from the same component as x.

The program of $\mathcal{S}_5$ consists of the following rules (one for each $a \in Y$) and their inverses:

$R(a)$ $\quad [x \to a^{-1}xa,\ p_1' \to p_0',\ q_1'r_1's_1't_1'\bar{p}_0 \to \delta^{-1}q_0'r_0's_0't_0'\bar{p}_1\delta,\ \bar{q}_0\bar{r}_0\bar{s}_0\bar{t}_0 \to \bar{q}_1\bar{r}_1\bar{s}_1\bar{t}_1]$

The machine $\mathcal{S}_6$ has the same hardware as $\mathcal{S}_5$ and just one rule:

$[p_0' \to p_1,\ q_0'r_0's_0't_0'\bar{p}_1' \to q_1r_1s_1t_1\bar{p}_0,\ \bar{q}_1'\bar{r}_1'\bar{s}_1'\bar{t}_1' \to \bar{q}_0\bar{r}_0\bar{s}_0\bar{t}_0]$

The machine $\mathcal{S}_7$ has the same hardware as $\mathcal{S}_5$ and $\mathcal{S}_6$ and one rule:

$[Ex \to Ex_4,\ p_1q_1r_1s_1t_1\bar{p}_0 \to p_4q_4r_4s_4t_4\bar{p}_4,\ \bar{q}_0\bar{r}_0\bar{s}_0\bar{t}_0 \to \bar{q}_4\bar{r}_4\bar{s}_4\bar{t}_4]$

It is clear that every nontrivial reduced computation of $\mathcal{S}_6$ or $\mathcal{S}_7$ has length 2.

The computations of $\mathcal{S}_5$ are more complicated but also easy to describe.

The proof of the following lemma is similar to the proofs of the previous lemmas and is left to the reader.

We shall need one more definition. Let $\langle Y\rangle$ be the free group with free basis Y and let $\langle 1\rangle$ be the additive group of integers. Consider the homomorphism from $\langle Y\rangle$ to $\langle 1\rangle$ which takes every element from Y to 1. Then the image of a word w from $\langle Y\rangle$ under this homomorphism will be denoted by $||w||$. Thus $||w||$ is the algebraic sum of the degrees of the letters in w. For example $||ab^{-1}c|| = 1$.

Lemma 4.7 *The following statements hold.*

(i) Any computation of $\mathcal{S}_5$ has a history of the form

$$R(a_1)R(a_2)^{-1}R(a_3)...$$

or

$$R(a_1)^{-1}R(a_2)R(a_3)^{-1}...$$

where $a_i \in Y$. If $z = a_1a_2^{-1}...$ or $z = a_1^{-1}a_2...$, $a_i \in Y$, then the computation starting with a word W and having the history corresponding to z will be denoted by $C(z, W)$.

(ii) Every computation of $\mathcal{S}_5$ is semiproper.

(iii) Suppose that $W = EuxvFE'p\delta^nqrst\bar{p}\delta^m\bar{q}\bar{r}\bar{s}\bar{t}F'$ where $u, v \in (Y \cup Y^{-1})^*$ and either

(1) $(p,q,r,s,t) = (p'_1, q'_1, r'_1, s'_1, t'_1)$ *and* $(\bar p, \bar q, \bar r, \bar s, \bar t) = (\bar p_0, \bar q_0, \bar r_0, \bar s_0, \bar t_0)$

or

(0) $(p,q,r,s,t) = (p'_0, q'_0, r'_0, s'_0, t'_0)$ *and* $(\bar p, \bar q, \bar r, \bar s, \bar t) = (\bar p_1, \bar q_1, \bar r_1, \bar s_1, \bar t_1)$.

Let $C = C(z, W)$ *be a computation of* $\mathcal{S}_5$ *and* $W' = \tau C$. *Then*

$$W' = Euz^{-1}xzvFE'p\delta^{n-\epsilon}qrst\bar p\delta^{m+\epsilon}\bar q\bar r\bar s\bar t F'$$

and

(0,0) *if* W *contains* p'_0 *and* W' *contains* p'_0 *then* $z = a_k a_{k-1}^{-1} a_{k-2} \ldots a_1^{-1}$ *for some even number* k, $a_i \in Y$, $i = 1, \ldots, k$, $\epsilon = 0$. *Such* z *will be called words of type (0,0). Notice that* $||z|| = 0$

(1,0) *if* W *contains* p'_1 *and* W' *contains* p'_0 *then* $z = a_k a_{k-1}^{-1} \ldots a_1$ *for some odd number* k, $a_i \in Y$, $i = 1, \ldots, k$, $\epsilon = 1$. *Such* z *will be called words of type (1,0). Notice that* $||z|| = 1$.

(0,1) *if* W *contains* p'_0 *and* W' *contains* p'_1 *then* $z = a_k^{-1} a_{k-1} \ldots a_1^{-1}$ *for some odd number* k, $a_i \in Y$, $i = 1, \ldots, k$, $\epsilon = -1$. *Such* z *will be called words of type (0,1). Notice that* $||z|| = -1$.

(1,1) *if* W *contains* p'_1 *and* W' *contains* p'_1 *then* $z = a_k^{-1} a_{k-1} \ldots a_1$ *for some even number* k, $a_i \in Y$, $i = 1, \ldots, k$, $\epsilon = 0$. *Such* z *will be called words of type (1,1). Notice that* $||z|| = 0$.

Notice that in each of the four cases if z *is a word of the type* (i,j) *then* $||z|| = i - j = \epsilon$.

We shall need three simple facts about words of type (ℓ, ℓ'), $\ell, \ell' \in \{0,1\}$ introduced in the previous lemma.

Lemma 4.8 *Let* z_j, $j = 1, \ldots, k$, *be an irreducible non-empty word of type* (ℓ_j, ℓ'_j) *where* $\ell_j, \ell'_j \in \{0,1\}$ *such that* $\ell_{j+1} = 1 - \ell'_j$ *for every* $j = 1, \ldots, k-1$. *Then the product* $z_k z_{k-1} \ldots z_1$ *(in the free semigroup) is an irreducible word.*

Proof. Indeed, an easy inspection of the definition of the words of type (ℓ, ℓ'), $\ell, \ell' \in \{0,1\}$, gives that if z_1 is of type $(\ell, 0)$ and z_2 is of type $(1, \ell')$ then z_2 ends with a positive letter and z_1 starts with a positive letter. If, on the other hand, z_1 is of type $(\ell, 1)$ and z_2 is of the type $(0, \ell')$ then z_2 ends with a negative letter and z_1 starts with a negative letter. In both cases there are no cancellations in the product $z_2 z_1$. This immediately implies the statement of the lemma. □

Lemma 4.9 *Let* v *be a positive word which does not end with letter* $a \in Y$. *Then* va^{-1} *cannot be represented in the form* $z_k z_{k-1} \ldots z_1$ *where* z_j *is a word of type* (ℓ_j, ℓ'_j), $\ell_j, \ell'_j \in \{0,1\}$ *such that* $\ell_{j+1} = 1 - \ell'_j$ *for every* $j = 1, \ldots, k-1$, *and* z_1 *is of type* $(1,0)$.

Proof. Let $v = a_1a_2...a_m$ where $a_i \in Y$, $i = 1, ..., m$ and $a_m \neq a$. Suppose that $va^{-1} = z_kz_{k-1}...z_1$ where the z_j satisfy the conditions of our lemma. Since z_1 must be of type $(1,0)$, it cannot end with a negative letter. Therefore the product $z_kz_{k-1}...z_1$ must have cancellations. But this contradicts Lemma 4.8. $\Box$

Lemma 4.10 *Let $v = a_ma_{m-1}...a_1$ be a word from Y^+ with $a_j \in Y$. Suppose that $v = z_kz_{k-1}...z_1$ where z_j is a word of type (ℓ_j, ℓ'_j), $\ell_j, \ell'_j \in \{0,1\}$ such that $\ell_{j+1} = 1 - \ell'_j$ for every $j = 1, ..., k-1$. Then $k = m$ and $z_j = a_j$ for every $j = 1, ..., k$.*

Proof. Indeed, since the product $z_kz_{k-1}...z_1$ is reduced by Lemma 4.8, z_j contains no negative letters. But it immediately follows from the definition of words of type (ℓ, ℓ') that if z_j is not a one letter word, it contains a negative letter. Thus each z_j is a one letter word: $m = k$ and $z_j = a_j$ for every $j = 1, ..., m$. $\Box$

The machine $\mathcal{S}_8$ is a cycle of the machines $\mathcal{S}_4$, $\mathcal{S}_5$, $\bar{\mathcal{S}}_4$, $\mathcal{S}_6$ and $\mathcal{S}_7$.

More precisely, $\mathcal{S}_8$ has the same hardware as $\mathcal{S}_5$, $\mathcal{S}_6$ and $\mathcal{S}_7$ and its program is the union of the programs of $\mathcal{S}_4$, $\mathcal{S}_5$, $\bar{\mathcal{S}}_4$, $\mathcal{S}_6$ and $\mathcal{S}_7$.

An informal description of $\mathcal{S}_8$ is the following. Suppose that it starts with a word

$$W = EuxFE'p_1\delta^n q_1r_1s_1t_1\bar{p}_0\bar{q}_0\bar{r}_0\bar{s}_0\bar{t}_0F'.$$

where $n \geq 0$. In general the work of the machine can be complicated, so we describe the "ideal" computation assuming that u is a positive word and $n = |u|$. The next lemma will show that this is the only computation from $C\mathcal{S}_8(W, p_4)$. First $\mathcal{S}_8$ checks whether $n > 0$. If $n = 0$ then by our assumption $u = \emptyset$ and $\mathcal{S}_8$ uses rules of $\mathcal{S}_7$ and finishes with the word

$$W_4 = Ex_4FE'p_4q_4r_4s_4t_4\bar{p}_4\bar{q}_4\bar{r}_4\bar{s}_4\bar{t}_4F'.$$

If $n > 0$ then $\mathcal{S}_8$ uses rules of $\mathcal{S}_5$ to move x one letter toward E, replaces n by $n-1$ and inserts δ between $\bar{p}$ and $\bar{q}$. Then $\mathcal{S}_8$ uses rules of $\bar{\mathcal{S}}_4$ to check that the word between $\bar{p}$ and $\bar{q}$ is not empty (which is true), then it uses $\mathcal{S}_4$ again to check if $n-1$ is not zero. If $n-1 = 0$ then $|u| = 1$, and $\mathcal{S}_8$ uses $\mathcal{S}_7$ and gets the word

$$W_4 = Ex_4uFE'p_4q_4r_4s_4t_4\bar{p}_4\delta\bar{q}_4\bar{r}_4\bar{s}_4\bar{t}_4F'.$$

If $n-1 > 0$ then $\mathcal{S}_8$ uses $\mathcal{S}_5$ again and moves x one letter closer to E, replaces $n-1$ by $n-2$ and inserts another δ between $\bar{p}$ and $\bar{q}$. This process repeats until x reaches E (and all n letters δ move to the right of $\bar{p}$). After that $\mathcal{S}_8$ uses $\mathcal{S}_7$ and we obtain the word

$$W_4 = Ex_4uFE'p_4q_4r_4s_4t_4\bar{p}_4\delta^n\bar{q}_4\bar{r}_4\bar{s}_4\bar{t}_4F'.$$

One can see that this is certainly not the only computation which can start with W. For example, at the beginning $\mathcal{S}_8$ can apply $\mathcal{S}_6$ (backward) instead of $\mathcal{S}_4$. Also after $\mathcal{S}_5$, the machine can proceed with using $\mathcal{S}_4$ instead of $\bar{\mathcal{S}}_4$ (if the computation of $\mathcal{S}_5$ is of the type (1,1)), etc. A description of all computations of $\mathcal{S}_8$ starting with W is contained in the following lemma.

Lemma 4.11 *($\mathcal{S}_8$ tells positive words in the alphabet Y from almost positive words.) Let*

$$W = EuxFE'p_1\delta^n q_1r_1s_1t_1\bar{p}_0\bar{q}_0\bar{r}_0\bar{s}_0\bar{t}_0F',$$

$$W_4 = Ex_4uFE'p_4q_4r_4s_4t_4\bar{p}_4\delta^n\bar{q}_4\bar{r}_4\bar{s}_4\bar{t}_4F',$$

$n \geq 0$. Then

(i) Every computation C of $\mathcal{S}_8$ starting with W or W_4 is semiproper. The length of C does not exceed $O((|W|+|V|)^2)$ and the area does not exceed $O((|W|+|V|)^3)$ where V is the last word in the computation C.

(ii) The set $C\mathcal{S}_8(W, xF, s_1t_1)$ consists of one (trivial) computation.

(iii) If u is a positive word then the statement $\mathcal{S}_8(W, p_4)$ is true if and only if $n = |u|$. In this case $C\mathcal{S}_8(W, p_4)$ consists of one computation and $\tau C\mathcal{S}_8(W, p_4) = W_4$. The length of this computation is $O(n^2)$ and the area is $O(n^3)$.

(iv) If u has the form $u'a^{-1}$ where $a \in Y$ and u' is a positive words, and u' does not end with a, then for every n the statement $\mathcal{S}_8(W, p_4)$ is false.

Proof. As before, we shall describe all computations of our machine which begin with W.

Let C be a reduced computation of our machine starting with W. It is clear that C has the form

$$C_1C_2...C_N. \tag{20}$$

where C_i is a non-trivial computation of one of the machines $\mathcal{S}_4$, $\mathcal{S}_5$, $\bar{\mathcal{S}}_4$, $\mathcal{S}_6$ and $\mathcal{S}_7$. Consecutive blocks should correspond to different machines.

It is easy to see that only the last block can be a computation of $\mathcal{S}_7$.

Let U_i be the first word in the block C_i, $i = 1, ..., N$. Then there should be rules of at least two machines among $\mathcal{S}_4, \mathcal{S}_5, \bar{\mathcal{S}}_4, \mathcal{S}_6, \mathcal{S}_7$ which are applicable to each of these words U_i. Easy inspection (checking all 10 possible pairs of machines) shows that U_i must have the form

$$Ev_ixw_iFE'p\delta^{n_i}qrst\bar{p}\delta^{m_i}\bar{q}\bar{r}\bar{s}\bar{t}F'$$

where $v_i, w_i \in (Y \cup Y^{-1})^*$, n_i, m_i are integers, and the word $pqrst\bar{p}\bar{q}\bar{r}\bar{s}\bar{t}$, which will be denoted by $\chi(U_i)$, is equal to one of the following words:

$P_1 = p_1q_1r_1s_1t_1\bar{p}_0\bar{q}_0\bar{r}_0\bar{s}_0\bar{t}_0$.

$P_2 = p'_1q'_1r'_1s'_1t'_1\bar{p}_0\bar{q}_0\bar{r}_0\bar{s}_0\bar{t}_0$.

$P_3 = p'_0q'_0r'_0s'_0t'_0\bar{p}_1\bar{q}_1\bar{r}_1\bar{s}_1\bar{t}_1$.

$P_4 = p'_0q'_0r'_0s'_0t'_0\bar{p}'_1\bar{q}'_1\bar{r}'_1\bar{s}'_1\bar{t}'_1$.

$P_5 = p_4q_4r_4s_4t_4\bar{p}_4\bar{q}_4\bar{r}_4\bar{s}_4\bar{t}_4$.

We are going to prove the following statements by induction on $i = 1, ..., N$. All of them are trivial for $i = 1$.

Fact 1. $n_i \geq 0$.

Fact 2. $m_i \geq 0$.

Fact 3. $n_i + m_i = n$.

Fact 4. $w_i = z_k z_{k-1} ... z_1$, $v_i = uw_i^{-1}$ where each word z_j $(j = 1, ..., k)$ is of type (ℓ_j, ℓ'_j) and these types satisfy the following property: $\ell_{j+1} = 1 - \ell'_j$, $j = 1, ..., k-1$. The number $k = k(i)$ is the number of blocks C_j, $j < i$, which are computations of $\mathcal{S}_5$. If C_N is a computation of $\mathcal{S}_7$ then the words z_1, z_k are of type $(1, 0)$.

Suppose that all these fact have been proven for all numbers $\leq i$.

Proof of Fact 1. If C_i is a computation of one of the machines $\mathcal{S}_4$, $\bar{\mathcal{S}}_4$, $\mathcal{S}_6$ then $n_{i+1} = n_i$ (for $\mathcal{S}_4$ and $\bar{\mathcal{S}}_4$ this follows from Lemma 4.6, for $\mathcal{S}_6$ it is obvious), so $n_{i+1} \geq 0$ as desired. Thus the only non-trivial case is when C_i is a computation of $\mathcal{S}_5$. It is easy to see that in this case $\chi(U_i) = P_2$ or $\chi(U_i) = P_3$.

Suppose first that $\chi(U_i) = P_2$. There are only two machines from the list $\{\mathcal{S}_4, \mathcal{S}_5, \bar{\mathcal{S}}_4, \mathcal{S}_6\}$ which have rules applicable to words with $\chi = P_2$. These are $\mathcal{S}_5$ and $\mathcal{S}_4$. Therefore C_{i-1} is a computation of $\mathcal{S}_4$. By Lemma 4.6 then $\chi(U_{i-1}) = P_1$ and $n_{i-1} = n_i > 0$. By Lemma 4.7, $n_{i+1} = n_i - \epsilon$ where $\epsilon \in \{0, 1\}$. Therefore $n_{i+1} \geq 0$.

Suppose now that $\chi(U_i) = P_3$. By Lemma 4.7 then $n_{i+1} = n_i - \epsilon$ where $\epsilon \in \{0, -1\}$. Since $n_i \geq 0$ by assumption, $n_{i+1} \geq 0$. This proves Fact 1. //

Proof of Fact 2. As in the proof of Fact 1, we can assume that C_i is a computation of $\mathcal{S}_5$. Again $\chi(U_i) = P_2$ or $\chi(U_i) = P_3$.

Suppose that $\chi(U_i) = P_2$. Then by Lemma 4.7, $m_{i+1} = m_i + \epsilon$ where $\epsilon \in \{1, 0\}$, so $m_{i+1} \geq 0$.

Now suppose that $\chi(U_i) = P_3$. There are only two machines in our list which are applicable to words with $\chi = P_3$. These are $\bar{\mathcal{S}}_4$ and $\mathcal{S}_5$. Therefore C_{i-1} is a computation of $\bar{\mathcal{S}}_4$. Then by Lemma 4.6, $\chi(U_{i-1}) = P_4$ and $m_{i-1} = m_i > 0$. Since by Lemma 4.7, $m_{i+1} = m_i + \epsilon$ where $\epsilon \in \{0, -1\}$, we can conclude that $m_{i+1} \geq 0$. This proves Fact 2.

Fact 3 is obvious because during the computation of $\mathcal{S}_8$ the total number of occurrences of δ stays the same.

Proof of Fact 4. Machines $\mathcal{S}_4$, $\bar{\mathcal{S}}_4$, $\mathcal{S}_6$ do not change v_i and w_i. By Lemma 4.7, $\mathcal{S}_5$ replaces x by $z^{-1}xz$ for some word $z \in (Y \cup Y^{-1})^*$ of type (ℓ, ℓ') where $\ell, \ell' \in \{0, 1\}$. This shows that for every i, $w_i = z_k z_{k-1} ... z_1$, z_j is of type (ℓ_j, ℓ'_j), and $u = v_i w_i$, where $k = k(i)$ is the number of computations of the machine $\mathcal{S}_5$ among the blocks $C_1, ..., C_{i-1}$. Let us prove that $\ell_{j+1} = 1 - \ell'_j$ for $j = 1, ..., k(i) - 1$.

Again we assume that this statement is proved for all numbers $\leq i$. Since only $\mathcal{S}_5$ can change v_i or w_i, we can assume that C_i is a computation of $\mathcal{S}_5$.

Let $w_i = z_k z_{k-1} ... z_1$ be the above representation of w_i.

Suppose that $\chi(U_i) = P_2$. Then by Lemma 4.7, $v_{i+1} = v_i z^{-1}$, $w_{i+1} = zw_i$ where z is of type $(1, h)$ for some $h \in \{0, 1\}$. We need to prove that $\ell'_k = 0$, that is z_k is of type $(\ell_k, 0)$. Without loss of generality we can assume that $k(i) > 0$.

As we mentioned in the proof of Fact 2, in this case C_{i-1} is a computation of $\mathcal{S}_4$. Since $n_i \geq 0$ (by Fact 1), we can apply Lemma 4.6 and conclude that $\chi(U_{i-1}) = P_1$ and $n_i > 0$. Since $k(i) > 0$, $i - 1 > 1$. There are only two machines applicable to U_{i-1}. These are $\mathcal{S}_4$ and $\mathcal{S}_6$. Therefore C_{i-2} is a computation of $\mathcal{S}_6$. Therefore $\chi(U_{i-2}) = P_4$. Since $P_4 \neq P_1$,

$i-2>1$. The block C_{i-3} must be a computation of $\bar{\mathcal{S}}_4$ because $\bar{\mathcal{S}}_4$ and $\mathcal{S}_6$ are the only machines in our list which have rules applicable to words with $\chi = P_4$. Since $m_i \geq 0$, we can apply Lemma 4.6 and conclude that $\chi(U_{i-3}) = P_3$. This implies that $i-3>1$ and C_{i-4} is a computation of $\mathcal{S}_5$. Since C_{i-4} is the last computation of $\mathcal{S}_5$ among the blocks C_j, $j<i$, by our assumption, C_{i-4} replaced x with $z_k^{-1}xz_k$ where $k=k(i)$. Since $U_{i-3} = \tau C_{i-4}$ and $\chi(U_{i-3}) = P_3$, by Lemma 4.7, z_k has type $(\ell, 0)$ for some ℓ. Therefore $\ell'_k = 0$ as desired.

Now suppose that $\chi(U_i) = P_3$. In this case by Lemma 4.7, C_i replaces x by $z^{-1}xz$ where z has type $(0,\ell')$ for some $\ell' \in \{0,1\}$. We need to prove that $\ell'_k = 1$, that is z_k is of type $(\ell_k, 1)$.

Arguing in the same way as before, we conclude that C_{i-1} is a computation of $\bar{\mathcal{S}}_4$, $\chi(U_{i-1}) = P_4$; C_{i-2} is a computation of $\mathcal{S}_6$, $\chi(U_{i-2}) = P_1$; C_{i-3} is a computation of $\mathcal{S}_1$, $\chi(U_{i-3}) = P_2$; C_{i-4} is a computation of $\mathcal{S}_5$. By our assumptions C_{i-4} replaces x by $z_k^{-1}xz_k$. Since $U_{i-3} = \tau C_{i-4}$ and $\chi(U_{i-3})$, by Lemma 4.7, the type of z_k is $(\ell_k, 1)$ for some ℓ_k, as desired.

Now let us prove that if C_N is a computation of $\mathcal{S}_7$ then z_1 is of type $(1,0)$. We have established that if C_i is the first block in (20) which is a computation of $\mathcal{S}_5$ then C_i replaces x with $z_1^{-1}xz_1$. We have also proved that $\chi(U_i) = P_2$ or $\chi(U_i) = P_3$, so $i>1$.

If $\chi(U_i) = P_2$ then C_{i-1} is a computation of $\mathcal{S}_4$, $\chi(U_{i-1}) = P_1$. If $i-1>1$ then C_{i-2} is a computation of $\mathcal{S}_6$, $\chi(U_{i-2}) = P_4$ so $i-2>1$. Then C_{i-3} is a computation of $\bar{\mathcal{S}}_4$, so $U_{i-3} = P_3$ and $i-3>1$. Then U_{i-4} must be a computation of $\mathcal{S}_5$. This contradicts the fact that C_i is the first block in (20) which is a computation of $\mathcal{S}_5$. Thus $i-1=1$ that is $i=2$. Therefore $C_{i-1} = C_1$ is a computation of $\mathcal{S}_4$. Therefore by Lemma 4.6,

$$U_2 = EuxFE'p'_1\delta^n q'_1 r'_1 s'_1 t'_1 \bar{p}_0\bar{q}_0\bar{r}_0\bar{s}_0\bar{t}_0F'.$$

and $n>0$.

Suppose that $\chi(U_3) = P_2$. Then by Lemma 4.7 z_1 has type (1,1) and

$$U_3 = Euz_1^{-1}xz_1FE'p'_1\delta^n q'_1 r'_1 s'_1 t'_1 \bar{p}_0\bar{q}_0\bar{r}_0\bar{s}_0\bar{t}_0F'.$$

Therefore C_3 is a computation of $\mathcal{S}_4$ (it cannot be a computation of $\mathcal{S}_7$), so $N>3$. This implies that

$$U_4 = Euz_1^{-1}xz_1FE'p_1\delta^n q_1 r_1 s_1 t_1 \bar{p}_0\bar{q}_0\bar{r}_0\bar{s}_0\bar{t}_0F'.$$

Since $n>0$, C_4 cannot be a computation of $\mathcal{S}_7$. So $N>4$ and C_4 is a computation of $\mathcal{S}_6$ and

$$U_5 = Euz_1^{-1}xz_1FE'p'_0\delta^n q'_0 r'_0 s'_0 t'_0 \bar{p}'_1\bar{q}'_1\bar{r}'_1\bar{s}'_1\bar{t}'_1F'.$$

Hence C_5 should be a computation of $\bar{\mathcal{S}}_4$ and U_6 should be equal to

$$Euz_1^{-1}xz_1FE'p'_0\delta^n q'_0 r'_0 s'_0 t'_0 \bar{p}_1\bar{q}_1\bar{r}_1\bar{s}_1\bar{t}_1F'.$$

But this is impossible by Lemma 4.6 since $m_6 = 0$. This means that $\chi(U_3) = P_3$, so by Lemma 4.7 z_1 has type (1,0) as desired.

Now suppose that $\chi(U_i) = P_3$. Then C_{i-1} is a computation of $\bar{\mathcal{S}}_4$. Then $\chi(U_{i-1}) = P_4$ and $m_i = m_{i-1} > 0$ (by Lemma 4.6). Then $i-1>1$ and C_{i-2} is a computation of $\mathcal{S}_6$.

This implies that $\chi(U_{i-2}) = P_1$, $m_{i-2} = m_{i-1} > 0$. Since $m_{i-2} > 0$, $i-2$ cannot be equal to 1 (since $U_{i-2} \neq U_1 = W$). Therefore C_{i-3} exists and is a computation of $\mathcal{S}_4$. Then $\chi(U_{i-3}) = P_2$ and $i-3 > 1$. Therefore C_{i-4} exists and is a computation of $\mathcal{S}_5$. This contradicts the assumption that C_i is the first block in (20) which is a computation of $\mathcal{S}_5$.

It remains to prove that if C_N is a computation of $\mathcal{S}_7$ then z_k has type $(1,0)$. We know that if C_i is the last block in (20) which is a computation of $\mathcal{S}_5$ then C_i replaces x by $z_k^{-1}xz_k$. As before, $\chi(U_i)$ is either P_2 or P_3.

Suppose first that $\chi(U_i) = P_2$. Then C_{i-1} is a computation of $\mathcal{S}_4$. This implies (by Lemma 4.6) that $n_i = n_{i-1} > 0$.

Suppose that $\chi(U_{i+1}) = P_2$. Then z_k is of type (1,1) and so $n_{i+1} = n_i > 0$. Furthermore, C_{i+1} is a computation of $\mathcal{S}_4$, so $\chi(U_{i+2}) = P_1$ and $n_{i+2} = n_{i+1} > 0$. Since $n_{i+2} > 0$, C_{i+3} cannot be a computation of $\mathcal{S}_7$. Therefore it must be a computation of $\mathcal{S}_6$. Then C_{i+4} must be a computation of $\bar{\mathcal{S}}_4$ and C_{i+5} must be a computation of $\mathcal{S}_5$ (both $i+4$ and $i+5$ are of course smaller than N, since C_N is a computation of $\mathcal{S}_7$). But this contradicts the fact that C_i is the last block in (20) which is a computation of $\mathcal{S}_5$. Therefore $\chi(U_{i+1}) = P_3$, so by Lemma 4.7, z_k has type $(1,0)$ as desired.

Now suppose that $\chi(U_i) = P_3$. Then again $\chi(U_{i+1}) = P_2$ or $\chi(U_{i+1}) = P_3$. We shall show that both possibilities lead to contradictions.

Suppose first that $\chi(U_{i+1}) = P_2$. Then C_{i-1} is a computation of $\mathcal{S}_4$, $\chi(U_{i+2}) = P_1$ and $n_{i+2} > 0$ (Lemma 4.6). Therefore C_{i+3} is not a computation of $\mathcal{S}_7$. Hence C_{i+3} is a computation of $\mathcal{S}_6$, C_{i+4} is a computation of $\bar{\mathcal{S}}_4$ and C_{i+5} is a computation of $\mathcal{S}_5$, a contradiction.

Finally let $\chi(U_{i+1}) = P_3$. Then by Lemma 4.7 the type of z_k is $(0,0)$. Since C_{i-1} is a computation of $\mathcal{S}_4$, $n_i > 0$ by Lemma 4.6. Since the type of z_k is (0,0) then by Lemma 4.7, $n_{i+1} = n_i > 0$. Then C_{i+1}, C_{i+2} are computations of, respectively, $\bar{\mathcal{S}}_4$ and $\mathcal{S}_6$. By Lemma 4.6, $n_{i+2} = n_{i+3} = n_{i+1} > 0$ and $\chi(U_{i+3}) = P_1$. Since $n_{i+3} > 0$, C_{i+4} is not a computation of $\mathcal{S}_7$, so it is a computation of $\mathcal{S}_4$. Then the next block, C_{i+5}, is a computation of $\mathcal{S}_5$, a contradiction.

This completes the proof of Fact 4.

Now let us complete the proof of the lemma.

Let C be a computation starting with W which has the form (20). Let again U_i be the first word in C_i. By Facts 1 – 4 that we have proved,

$$U_i = Ev_ixw_iFE'p\delta^{n_i}qrst\bar{p}\delta^{m_i}\bar{q}\bar{r}\bar{s}\bar{t}F'$$

where $v_i, w_i \in (Y \cup Y^{-1})^*$, n_i, m_i are integers, the word $\chi(U_i) = pqrst\bar{p}\bar{q}\bar{r}\bar{s}\bar{t}$ is equal to one of the words P_1, P_2, P_3, P_4, P_5, $w_i = z_kz_{k-1}...z_1$ where each z_j is of type (ℓ_j, ℓ'_j) for some ℓ_j, ℓ'_j, $v_i = uw_i^{-1}$, $n_i \geq 0$, $m_i \geq 0$.

Then by Lemmas 4.6 and 4.7 all blocks C_i are semiproper. Moreover, all C_i, $i \leq N-1$, which are computations of the machines $\mathcal{S}_4$, $\bar{\mathcal{S}}_4$, $\mathcal{S}_6$ and $\mathcal{S}_7$ are proper: no negative letters can be inserted during these computations because in the initial and final words in these computations, the parts of the words which can be modified by the rules of these machines are positive (since $n_i \geq 0, m_i \geq 0$). All negative letters inserted during computations of $\mathcal{S}_5$ between E and F belong to the subwords z_j. But by Lemma 4.8 the word $z_kz_{k-1}...z_1$ is reduced, so no negative letter inserted between E and F during one of the computations of $\mathcal{S}_5$ can be cancelled during another computation of $\mathcal{S}_5$. We also already know that the blocks C_i corresponding to $\mathcal{S}_5$ did not insert negative letters (δ's) between E' and F' because these blocks are semiproper (Lemma 4.7) and n_i and m_i are always non-negative.

This proves the first part of (i): every computation of $\mathcal{S}_8$ starting with W is semiproper. Let N_1 be the number of computations of $\mathcal{S}_5$ among blocks in (20). Then our analysis shows that $N \leq 4N_1$ and N_1 is equal to the number of subwords z_i. Thus $N_1 \leq |\tau C|$. If C_i is a computation of $\mathcal{S}_5$ then the length of C_i is bounded by the length of the corresponding word z_i. Therefore it is bounded by the length of $|\tau C|$. The area of this block is bounded by $|\tau C|^2$. Thus the sum of lengths of these blocks is bounded by $|\tau C|^2$ and the total area of these blocks is bounded by $|\tau C|^3$. By Lemma 4.6 the length of each of the other blocks is bounded by $O(|\tau C| + |W|)$ (notice that the lengths of U_i do not decrease when i goes from 1 to N). And again the area of each of the other blocks does not exceed $O((|\tau C| + |W|)^2)$. This implies that the length of C does not exceed $O((|\tau C| + |W|)^2)$ and the area does not exceed $O((|\tau C| + |W|)^3)$. This proves the second part of (i).

Let C be a computation in $C\mathcal{S}_8(W, xF, s_1t_1)$. Let $W' = \tau C$. By Fact 4 and Lemma 4.8 there are no blocks in the representation (20) of C which are computations of $\mathcal{S}_5$ (otherwise the subword in W' between x and F would not be empty).

This implies that either C has just one block which is a computation of $\mathcal{S}_4$ or $\mathcal{S}_6$, or it has two blocks C_1C_2 where C_1 is a computation of $\mathcal{S}_6$ and C_2 is a computation of $\bar{\mathcal{S}}_4$.

The latter case is impossible because then W' would not have the subword s_1t_1 ($\mathcal{S}_6$ changes s_1 to s_0' and $\bar{\mathcal{S}}_4$ does not touch state letters without a "$-$").

Any nontrivial reduced computation of $\mathcal{S}_6$ consists of two words, and there are no nontrivial computations in $C\mathcal{S}_6(W, s_1t_1)$. Therefore C cannot be a computation of $\mathcal{S}_6$.

Finally by Lemma 4.6, every computation of $C\mathcal{S}_4(W, s_1t_1)$ is trivial. Thus C is not a computation of $\mathcal{S}_4$. All cases have been considered, so $C\mathcal{S}_8(W, xF, s_1t_1)$ consists of one (trivial) computation. This proves (ii).

Suppose that u is a positive word and $n = |u|$. Then there exists a computation of $\mathcal{S}_8$ which starts with W and ends with W_4. This computation has the following representation as a sequence of blocks:

$$C_{4,1}C_{5,1}C_{\bar{4},1}C_{6,1}C_{4,2}...C_{4,n}C_{5,n}C_{\bar{4},n}C_{6,n}C_{7,1} \tag{21}$$

where $C_{i,j}$ is a computation of $\mathcal{S}_i$ ($i = 4, 5, 6, 7,\ j = 1, ..., n$) and $C_{\bar{4},j}$ is a computation of $\bar{\mathcal{S}}_4$. Every computation of $\mathcal{S}_5$ in this sequence consists of two words, so all z_i are one letter words.

Now suppose that there exists a computation C in $C\mathcal{S}_8(W, p_4)$. Let $W' = \tau C\mathcal{S}_8(W, p_4)$. By Facts 3 and 4, $W' = W_4$. By Fact 4, $u = z_k z_{k-1} ... z_1$ where the z_i's satisfy the properties listed in Fact 4. By Lemma 4.10, since u is a positive word, every z_i is a one letter word. Therefore by Fact 4, every block in (20) which is a computation of $\mathcal{S}_5$ consists of two words. Our description of computations of $\mathcal{S}_8$ starting with W (see proofs of Facts 1–4) show that the computation must have the form

$$C_{4,1}C_{5,1}C_{\bar{4},1}C_{6,1}C_{4,2}...C_{4,k}C_{5,k}C_{\bar{4},k}C_{6,k}C_{7,1}$$

where $k = |u|$. Each block $C_{5,j}$ consists of an application of a rule of the form $R(a)$ which subtracts 1 from n, the number of δ's between p and t (here we omit indices in p and t).

Other blocks do not touch this number. Since there is no δ between p and t in the last word in $C_{6,k}$ (otherwise we would not be able to apply the rule of $\mathcal{S}_7$ to this word), k must be equal to n.

Thus we proved (iii): if u is positive then $C\mathcal{S}_8(W, p_4)$ is not empty if and only if $n = |u|$; in this case $C\mathcal{S}_8(W, p_4)$ consists of one computation (21). The length of this computation can be easily computed with the help of Lemma 4.6 (iii). It is equal to $O(n + (n-1) + ... + 1) = O(n^2)$.

Statement (iv) follows from Fact 4 and Lemma 4.9. Indeed, by Fact 4, if $C\mathcal{S}_8(W, p_4)$ is not empty then $u = z_k z_{k-1} ... z_1$ where the z_j's satisfy the properties listed in Fact 4. By Lemma 4.9 such a representation of u is impossible if $u = u'a^{-1}$ where $a \in Y$, and u' is a positive word which does not end with a.

This completes the proof of the Lemma. $\square$.

The machine $\mathcal{S}_9$ is obtained from $\mathcal{S}_8$ in the same way $\mathcal{S}_4$ was obtained from $\mathcal{S}_3$. Let $\mathcal{S}_8'$ be a copy of $\mathcal{S}_8$ which is obtained by adding $\hat{}$ to all state letters except E, F, E', F' and those state letters which have index 4. Let $\mathcal{S}_9$ be the concatenation of $\mathcal{S}_8$ and $\mathcal{S}_8'$. Namely, the tape alphabet vector $Y(9)$ of $\mathcal{S}_9$ is the same as the tape alphabet of $\mathcal{S}_8$, the state alphabet vector $Q(9)$ is the union of state alphabet vectors of $\mathcal{S}_8$ and $\mathcal{S}_8'$, and the program of $\mathcal{S}_9$ is the union of the programs of $\mathcal{S}_8$ and $\mathcal{S}_8'$.

The following Lemma is similar to Lemma 4.6.

Lemma 4.12 *($\mathcal{S}_9$ tells positive words in the alphabet Y from almost positive words and returns the state letters to their original positions.) Let*

$$W = EuxFE'p_1\delta^n q_1 r_1 s_1 t_1 \bar{p}_0 \bar{q}_0 \bar{r}_0 \bar{s}_0 \bar{t}_0 F'$$

where u is either a positive word or a reduced word of the form $u'a^{-1}$ where u' is a positive word and a is a letter. Let

$$\hat{W} = Eu\hat{x}FE'\hat{p}_1\delta^n \hat{q}_1 \hat{r}_1 \hat{s}_1 \hat{t}_1 \hat{\bar{p}}_1 \hat{\bar{q}}_0 \hat{\bar{r}}_0 \hat{\bar{s}}_0 \hat{\bar{t}}_0 F'$$

$n \geq 0$. Then

(i) *Every computation C of $\mathcal{S}_9$ starting with W or $\hat{W}$ is semiproper. The length of C does not exceed $O((|W| + |V|)^2)$ and the area does not exceed $O((|W| + |V|)^3)$ where V is the last word in the computation.*

(ii) *Each of the sets $C\mathcal{S}_9(W, xF, s_1 t_1)$ and $C\mathcal{S}_9(\hat{W}, \hat{x}F, \hat{s}_1 \hat{t}_1)$ consists of one (trivial) computation.*

(iii) *The statement $\mathcal{S}_9(W, \hat{x})$ (statement $\mathcal{S}_9(\hat{W}, x)$) is true if and only if u is a positive word and $n = |u|$. In this case $C\mathcal{S}_9(W, \hat{x})$ (resp. $C\mathcal{S}_9(\hat{W}, x)$) consists of one computation and $\tau C\mathcal{S}_9(W, \hat{x}) = \hat{W}$ (resp. $\tau C\mathcal{S}_9(\hat{W}, x) = W$). The length of this computation is $O(n^2)$.*

(iv) *If u has the form $u'a^{-1}$ where $a \in Y$ and u' is a positive word, and u' does not end with a, then for every n the statement $\mathcal{S}_9(W, \hat{x})$ and the statement $\mathcal{S}_9(\hat{W}, x)$ are false.*

Proof. Every computation C of $\mathcal{S}_9$ starting with W can be represented in the form $C_1C_2C_3$.... where C_i is a computation of one of the machines $\mathcal{S}_8$ or $\mathcal{S}'_8$, neighboring blocks are computations of different machines, and each block is non-trivial.

Since C starts with the word W, C_1 must be a computation of the machine $\mathcal{S}_8$. If $C = C_1$ then C is a computation of $\mathcal{S}_8$ and we can apply Lemma 4.11. It is clear that in this case τC cannot contain $\hat{x}$.

If $C \neq C_1$ then C_2 is a computation of $\mathcal{S}'_8$. Therefore C_2 starts with a word U containing p_4. This word is the end word of C_1. So statement $\mathcal{S}_8(W, p_4)$ holds and by Lemma 4.11 (iii), (iv) u is positive and $n = |u|$. Now by Lemma 4.11 (i) and Lemma 4.2 the computation C_1 is proper (it is semiproper and the end word is positive) and C_2 is semiproper.

Suppose that C_3 exists. Then it must be a computation of $\mathcal{S}_8$. Therefore it must start with a word containing p_4. By Lemma 4.11(ii) then C_2 is trivial, a contradiction.

Thus C consists of one or two blocks: $C = C_1$ or $C = C_1C_2$. In both cases C is semiproper. The computation C can contain two blocks if and only if u is positive and $n = |u|$.

Now all statements of the lemma follow immediately from this description and Lemmas 4.11, 4.1 and 4.2. $\square$

The admissible words of the S-machine $\mathcal{S}_\alpha$ have the following form:

$$E\alpha^k x\alpha^n F$$

where x is one of the letters x, x_1, x_2. The rules of $\mathcal{S}_\alpha$ are the following:

(α1) $x \to \alpha^{-1}x\alpha$,

(α_2) $Ex \to Ex_1$,

(α_3) $x_1 \to \alpha x_1 \alpha^{-1}$,

(α_4) $x_1F \to x_2F$.

Lemma 4.13 *($\mathcal{S}_\alpha$ moves x from F to E and back). Let $W = E\alpha^n xF$, $W_2 = E\alpha^n x_2 F$, $n \geq 0$. Then:*

1. *Every reduced computation C starting with W or W_2 is semiproper. The length of C does not exceed $O(|W| + |V|)$ and the area does not exceed $O((|W| + |V|)^2)$ where $V = \tau C$.*

2. *Each of the sets $C\mathcal{S}_\alpha(W, xF)$ and $C\mathcal{S}_\alpha(W_2, x_2)$ consists of one (trivial) computation.*

3. *$C\mathcal{S}_\alpha(W, x_2)$ consists of exactly one computation with the history word*

$$(\alpha_1)^n(\alpha_2)(\alpha_3)^n(\alpha_4)$$

and $\tau C\mathcal{S}_\alpha(W, x_2) = W_2$, and Ex_1 is a subword of a word in this computation.

Proof. Indeed it is easy to see that the history h of every computation C starting with W is a prefix of the following word

$$(\alpha 1)^k(\alpha 2)(\alpha 3)^\ell(\alpha 4)$$

for some numbers k and ℓ. This and Lemma 4.1 immediately imply statements 1 and 2 of the lemma.

If the history h contains $(\alpha 2)$ then $k = n$ because every application of $(\alpha 1)$ moves x one letter closer to E. Similarly if h contains $(\alpha 4)$ then it must contain $(\alpha 2)$ and ℓ must be equal to n. This implies statement 3. $\Box$

The machine $\mathcal{S}_\omega$ is similar to $\mathcal{S}_\alpha$.

The admissible words of the S-machine $\mathcal{S}_\omega$ have the following form:

$$E'\omega^k x'\omega^n F$$

where x' is one of the letters x', x'_1, x'_2. The rules of $\mathcal{S}_\omega$ are the following:

$(\omega 1)$ $x' \to \omega x \omega^{-1}$,

(ω_2) $x'F' \to x'_1 F'$,

(ω_3) $x'_1 \to \omega^{-1} x'_1 \omega$,

(ω_4) $E'x'_1 \to E'x'_2$.

The following lemma is similar to Lemma 4.13, so its proof is omitted.

Lemma 4.14 *($\mathcal{S}_\omega$ moves x' from E' to F' and back). Let $W = E'x\omega^n F'$, $W_2 = E'x_2\omega^n F'$, $n \geq 0$. Then:*

1. *Every reduced computation C starting with W or W_2 is semiproper. The length of C does not exceed $O(|W| + |V|)$ and the area does not exceed $O((|W| + |V|)^2$ where $V = \tau C$.*

2. *Each of the sets $C\mathcal{S}_\omega(W, E'x')$ and $C\mathcal{S}_\alpha(W_2, x'_2)$ consists of one (trivial) computation.*

3. *$C\mathcal{S}_\omega(W, x'_2)$ consists of exactly one computation with the history word*

$$(\omega_1)^n(\omega_2)(\omega_3)^n(\omega_4)$$

and $\tau C\mathcal{S}_\omega(W, x'_2) = W_2$, and $x'_1 F'$ is a subword of a word in this computation.

Now we are ready to take any Turing machine

$$M = \langle X, Y, Q, \Theta, \vec{s}_1, \vec{s}_0 \rangle$$

satisfying the conditions of Lemma 3.1 and to construct an S-machine $\mathcal{S}(M)$ simulating M.

For every $q \in Q$ we denote the word $q\omega$ by F_q. Let us replace $q\omega$ in every command of M by F_q . After that we won't have q's in any command of M because of part 5 of Lemma 3.1. In order to have notation similar to that of the machine $\mathcal{S}_9$, we also denote the left marker on tape $\#i$ by E_i. This gives us a Turing machine M' such that the configurations of each tape of M' have the form E_iuF_q where u is a word in the alphabet of tape i, and every command or its inverse has one of the forms:

$$\{F_{q_1} \to F_{q'_1}, ..., aF_{q_i} \to F_{q'_i}, ..., F_{q_k} \to F_{q'_k}\} \tag{22}$$

where $a \in Y$ or

$$\{F_{q_1} \to F_{q'_1}, ..., E_iF_{q_i} \to E_iF_{q'_i}, ..., F_{q_k} \to F_{q'_k}\} \tag{23}$$

This machine recognizes the same language and has the same complexity functions as M, so we can assume that M itself has this form.

An admissible word of the S-machine $\mathcal{S}(M)$ is a product of three parts. The first part has the form

$$E(0)\alpha^{n_1}x(0)\alpha^{n_2}F(0)$$

The second part is a product of k words of the form

$$E(i)u_ix(i)v_iF(i)E'(i)p(i)\Delta_{i,1}q(i)\Delta_{i,2}r(i)\Delta_{i,3}s(i)\Delta_{i,4}t(i)\Delta_{i,5}$$
$$\bar{p}(i)\Delta_{i,6}\bar{q}(i)\Delta_{i,7}\bar{r}(i)\Delta_{i,8}\bar{s}(i)\Delta_{i,9}\bar{t}(i)\Delta_{i,10}F'(i)$$

$i = 1, ..., k$. The third part has the form

$$E'(k+1)\omega^{n'_1}x'(k+1)\omega^{n'_2}F'(k+1)$$

Here u_i, v_i are group words in the alphabet Y_i of tape i, and $\Delta_{i,j}$ is a power of δ. The letters

$$E(i), x(i), F(i), E'(i), p(i), q(i), r(i), s(i), t(i), \bar{p}(i), \bar{q}(i), \bar{r}(i), \bar{s}(i), \bar{t}(i), F'(i)$$

belong to disjoint sets of state letters which we shall denote, respectively, by

$$\mathbf{E}(i), \mathbf{X}(i), \mathbf{F}(i), \mathbf{E}'(i), \mathbf{P}(i), \mathbf{Q}(i), \mathbf{R}(i), \mathbf{S}(i), \bar{\mathbf{P}}(i), \bar{\mathbf{Q}}(i), \bar{\mathbf{R}}(i), \bar{\mathbf{S}}(i), \bar{\mathbf{T}}(i), \mathbf{F}'(i),$$

$i = 0, ..., k+1$. The description of these sets is below.

Since M is symmetric, for every command from Θ its inverse is in Θ. Let us call commands of the form (22) *positive*, the inverses of these commands will be called *negative*. We also choose one from each pair of mutually inverse commands of the form (23) and call it *positive*; the other command in this pair will be called *negative*. As we have seen before, every command in Θ is either positive or negative.

Now let us describe the set of state letters of $\mathcal{S}(M)$. First for every $i = 0, ..., k+1$ we include the letter $E(i)$ into $\mathbf{E}(i)$ and the letter $E'(i)$ into $\mathbf{E}'(i)$. Then for every state letter F on tape i we include F into $\mathbf{F}(i)$ and F' into $\mathbf{F}'(i)$. We also include the letters

$$x(i), p(i), q(i), r(i), s(i), t(i), \bar{p}(i), \bar{q}(i), \bar{r}(i), \bar{s}(i), \bar{t}(i)$$

into the corresponding sets $\mathbf{X}(i)$, $\mathbf{P}(i)$, $\mathbf{Q}(i)$, $\mathbf{R}(i)$, $\mathbf{S}(i)$, $\mathbf{T}(i)$, $\bar{\mathbf{P}}(i)$, $\bar{\mathbf{Q}}(i)$, $\bar{\mathbf{R}}(i)$, $\bar{\mathbf{S}}(i)$, $\bar{\mathbf{T}}(i)$, $(i = 1, ..., k)$. The letters that we just described will be called *standard*.

Now let τ be a positive command in Θ. Then for every $\gamma \in \{4, 9, \alpha, \omega\}$ and for each component $\mathbf{U}(i)$ of the vector of sets of state letters, we include the letter $U(i, \tau, \gamma)$ into $\mathbf{U}(i)$.

Suppose that τ has the form (22):

$$\tau = \{F_{q_1} \to F_{q'_1}, ..., aF_{q_i} \to F_{q'_i}, ..., F_{q_k} \to F_{q'_k}\}.$$

for some i from 1 to k. For each S-machine $\mathcal{S}_\gamma$ where $\gamma \in \{4, 9, \alpha, \omega\}$ we consider a copy $\mathcal{S}_\gamma(\tau)$ where every state letter z is replaced by $z(j, \tau, \gamma)$ where $j = i$ if $\gamma = 4, 9$, $j = 0$ if $\gamma = \alpha$ and $j = k+1$ if $\gamma = \omega$. We include these state letters into the corresponding sets $\mathbf{P}(i)$, $\mathbf{Q}(i)$, $\mathbf{R}(i)$, $\mathbf{S}(i)$, $\mathbf{T}(i)$, $\bar{\mathbf{P}}(i)$, $\bar{\mathbf{Q}}(i)$, $\bar{\mathbf{R}}(i)$, $\bar{\mathbf{S}}(i)$, $\bar{\mathbf{T}}(i)$ if $\gamma \in \{4, 9\}$, in $\mathbf{E}(0)$, $\mathbf{X}(0)$, $\mathbf{F}(0)$ if $\gamma = \alpha$ and into $\mathbf{E}(k+1)$, $\mathbf{X}(k+1)$, $\mathbf{F}(k+1)$ if $\gamma = \omega$.

The set of state letters that we just described is the set of all state letters of $\mathcal{S}(M)$.

Now let us describe the set of rules of $\mathcal{S}(M)$. It will consist of the rules of $\mathcal{S}_4(\tau)$, $\mathcal{S}_9(\tau)$, $\mathcal{S}_\alpha(\tau)$, $\mathcal{S}_\omega(\tau)$ for all τ of the form (22) plus the following connecting rules. First let τ be a command of the form (22):

$$\tau = \{F_{q_1} \to F_{q'_1}, ..., aF_{q_i} \to F_{q'_i}, ..., F_{q_k} \to F_{q'_k}\}.$$

Notice that τ determines the letter a and the index i.

The machine $\mathcal{S}(M)$ simulates this command as follows. First, using $\mathcal{S}_4(\tau)$, it checks whether the word between $E'(i)$ and $F'_{q_i}(i)$ is not empty. If it is empty, the execution cannot proceed to the next step. Otherwise the machine changes q_j to q'_j in the indices of the F's, inserts a^{-1} next to the left of $x(i)$, removes one δ in the word between $E'(i)$ and $F'_{q_i}(i)$, removes one α and removes one ω. Then it uses $\mathcal{S}_\alpha(\tau)$ and $\mathcal{S}_\omega(\tau)$ to move $x(0)$ from $F(0)$ to $E(0)$ and back and to move $x(k+1)$ from $E'(k+1)$ to $F'(k+1)$ and back (the purpose of these seemingly needless moves will be clear later). Finally it checks (using $\mathcal{S}_9(\tau)$) if after we inserted a^{-1}, the word between $E(i)$ and $F_{q'_i}(i)$ is positive. If it is positive, the machine gets ready to execute the next transition of Θ (by returning all state letters to their standard forms). Otherwise, the machine cannot proceed to the execution of the next transition from Θ.

Here is the description of the connecting rules in the case when τ has the form (22).

$R_4(\tau)$. This rule is applicable to an admissible word W if all state letters of W are standard, and W contains the subwords

$$x(j)F_{q_j}(j)E'(j)p(j)$$

and

$$q(j)r(j)s(j)t(j)F'_{q_j}(j)$$

for every $1 \leq j \leq k$, and also the subwords $x(0)F(0)$ and $E'(k+1)x'(k+1)$. It changes each standard state letter $z(j)$ to $z(j, \tau, 4)$. The meaning of this rule is simple: it turns on the machine $\mathcal{S}_4(\tau)$.

$R_{4,\alpha}(\tau)$. This rule is applicable when all state letters in an admissible word W have $(\tau,4)$ in their labels, and in addition W contains the subword $s_1'(i,\tau,4)t_1'(i,\tau,4)$ (that is when $\mathcal{S}_4(\tau)$ finishes its work). It

- changes q_j to q_j' in the indices of F $(j=1,...,k)$,
- replaces
$$x(0,\tau,4)F(0,\tau,4)$$
by
$$\alpha^{-1}x(0,\tau,\alpha)F(0,\tau,\alpha),$$
- replaces
$$E'(k+1,\tau,4)x(k+1,\tau,4)$$
by
$$E'(k+1,\tau,\alpha)x(k+1,\tau,\alpha)\omega^{-1},$$
- replaces
$$x(i,\tau,4)F_{q_i}(i,\tau,4)E'(i,\tau,4)p_1'(i,\tau,4)$$
by
$$a^{-1}x(i,\tau,\alpha)F_{q_i'}(i,\tau,\alpha)E'(i,\tau,\alpha)p_1(i,\tau,\alpha)\delta^{-1},$$
- replaces
$$q_1'(i,\tau,4)r_1'(i,\tau,4)s_1'(i,\tau,4)t_1'(i,\tau,4)$$
by
$$q_1(i,\tau,\alpha)r_1(i,\tau,\alpha)s_1(i,\tau,\alpha)t_1(i,\tau,\alpha)$$
- replaces $(\tau,4)$ by (τ,α) in all other state letters.

This rule simulates execution of the transition τ and turns on the machine $\mathcal{S}_\alpha(\tau)$.

$R_{\alpha,\omega}(\tau)$. This rule is applicable when all state letters in an admissible word have (τ,α) in their labels and $x_2(0,\tau,\alpha)$ occurs in this word. It changes (τ,α) to (τ,ω) in the labels of all state letters, and replaces $x_2(0,\tau,\alpha)$ by $x(0,\tau,\omega)$. This rule turns on the machine $\mathcal{S}_\omega(\tau)$.

$R_{\omega,9}(\tau)$ This rule is applicable when all state letters in an admissible word have (τ,ω) in their labels, and the letter $x_2(k+1,\tau,\omega)$ occurs in this word. It changes (τ,ω) to $(\tau,9)$ and replaces $x_2(k+1,\tau,\omega)$ by $x(\tau,9)$. This rule turns on the S-machine $\mathcal{S}_9$.

$R_9(\tau)$. This rule applies to an admissible word W when all state letters in W have $(\tau,9)$ in their labels, and W contains $\hat{x}(i,\tau,9)$, that is when $\mathcal{S}_9(\tau)$ ends its work. The rule removes "^" from all letters, removes $(\tau,9)$ and indices from all state letters, i.e. this rule returns the state letters to their standard form. The meaning of this rule is that it turns off $\mathcal{S}_9(\tau)$ and gets our machine ready to simulating the next transition from θ.

We shall consider the S-machines $R_4(\tau)$, $R_{4,\alpha}(\tau)$, $R_{\alpha,\omega}(\tau)$, $R_{\omega,9}(\tau)$, $R_9(\tau)$ whose hardware is the same as the hardware of $\mathcal{S}(M)$ and whose only rules are, respectively, $R_4(\tau)$, $R_{4,\alpha}(\tau)$, $R_{\alpha,\omega}(\tau)$, $R_{\omega,9}(\tau)$, $R_9(\tau)$. We shall call these S-machines *transition machines.*

Now let τ have the form (23),

$$\tau = \{F_{q_1} \to F_{q'_1}, ..., E_i F_{q_i} \to E_i F_{q'_i}, ..., F_{q_k} \to F_{q'_k}\}.$$

In this case the simulation is much easier. It consists of just one S-rule:

$$\begin{aligned} P(\tau) \ &= [F_{q_1}(1) \to F_{q'_1}(1), ..., \\ &E(i)x(i)F_{q_i}(i)E'(i)p(i)q(i)r(i)s(i)t(i)\bar{p}(i)\bar{q}(i)\bar{r}(i)\bar{s}(i)\bar{t}(i)F'_{q_i}(i) \\ &\to E(i)x(i)F_{q'_i}(i)E'(i)p(i)q(i)r(i)s(i)t(i)\bar{p}(i)\bar{q}(i)\bar{r}(i)\bar{s}(i)\bar{t}(i)F'_{q'_i}(i), \\ &..., F_{q_k}(k) \to F_{q'_k}(k)] \end{aligned}$$

For every configuration $c = (E_1 u_1 F_{q_1}, ..., E_k u_k F_{q_k})$ of the machine M let $\sigma(c)$ be the following admissible word of $\mathcal{S}(M)$:

$$\begin{array}{l} E(0)\alpha^n x(0)F(0) \\ E(1)u_1 x(1)F_{q_1}(1)E'(1)p(1)\delta^{||u_1||}q(1)r(1)s(1)t(1)\bar{p}(1)\bar{q}(1)\bar{r}(1)\bar{s}(1)\bar{t}(1)F'_{q_1}(1)... \\ E(k)u_k x(k)F_{q_k}(k)E'(k)p(k)\delta^{||u_k||}q(k)r(k)s(k)t(k)\bar{p}(k)\bar{q}(k)\bar{r}(k)\bar{s}(k)\bar{t}(k)F'_{q_k}(k) \\ E'(k+1)x(k+1)\omega^n F'(k+1) \end{array}$$

where $n = |u_1| + ... + |u_k|$. Notice that $|\sigma(c)| = 4|c| + 13k + 6$.

Conversely, if W is an admissible word for $\mathcal{S}(M)$, let $\mu(W)$ be the word obtained by removing

$$x, p, q, r, s, t, \bar{p}, \bar{q}, \bar{r}, \bar{s}, \bar{t}, \alpha, \omega, \delta, E(0), F(0)$$

and all E', F' with indices, replacing state letters by their standard forms (that is removing (τ, j) from their labels), and reducing the resulting word. It is clear that if W is positive then $\mu(W)$ is a configuration of the machine M. It is clear also that $\mu(\sigma(c)) = c$.

We shall call an admissible word W of $\mathcal{S}(M)$ *normal* if (using the notation in the definition of admissible words for $\mathcal{S}(M)$) for every i from 1 to k we have that $||u_i v_i|| = ||\Delta_{i,1}...\Delta_{i,10}||$ and if $n_1 + n_2 = n'_1 + n'_2 = \sum_{i=1}^k ||u_i v_i|| \geq 0$. Recall that $||u||$ is the algebraic sum of the degrees of all letters in W.

The following statement is obvious.

Lemma 4.15 *Let W be an admissible word for $\mathcal{S}(M)$. Then if W is positive and normal and one of the rules $R_4(\tau), R_9(\tau)^{-1}, P(\tau)$, $\tau \in \Theta$ is applicable to W then $W = \sigma(c)$ for some configuration c of the machine M.*

If a transition τ has the form (22): $\tau = \{F_{q_1} \to F_{q'_1}, ..., aF_{q_i} \to F_{q'_i}, ..., F_{q_k} \to F_{q'_k}\}$, then the subword between an $\mathbf{E}(i)$-letter and an $\mathbf{F}'(i)$-letter in an admissible word of $\mathcal{S}(M)$ will be called the *τ-part* of this word. The subword between an $\mathbf{E}'(i)$-letter and an $\mathbf{F}'(i)$-letter will be called the (τ, δ)-part, the subword between an $\mathbf{E}(i)$-letter and an $\mathbf{F}(i)$-letter will be called the (τ, M)-part, the subword between an $\mathbf{E}(0)$-letter and an $\mathbf{F}(0)$-letter will be called the α-part, and the subword between an $\mathbf{E}'(k+1)$-letter and an $\mathbf{F}'(k+1)$-letter will be called the ω-part. Notice that the work of $\mathcal{S}_4(\tau)$ affects only the (τ, δ) parts of admissible words and the work of $\mathcal{S}_9(\tau)$ affects only the τ-parts, the work of $\mathcal{S}_\alpha(\tau)$ affects only the α-parts and the work of $\mathcal{S}_\omega$ affects only the ω-parts. Therefore we can apply Lemmas 4.6 and 4.12 when we talk about $\mathcal{S}_4(\tau)$ and $\mathcal{S}_9(\tau)$ even though the hardware of, say, $\mathcal{S}_4(\tau)$ is "bigger" than the hardware of $\mathcal{S}_4$.

Let c_0 be the accept configuration of the machine M (all tapes are empty, the indices of the F's form the accept vector $\vec{s}_0$). Then we can define the generalized time function $d(n)$ and the area function $a(n)$ of the S-machine $\mathcal{S}(M)$. For every $n > 1$ we denote by $d(n)$ (resp. $a(n)$) the smallest number such that every admissible word W of length $\leq n$ for which the set of computations $C\mathcal{S}(M)(W, \sigma(c_0))$ is not empty, this set contains a computation of length (resp. area) $\leq d(n)$ (resp. $\leq a(n)$). The function $d(n)$ is called the *generalized time function* and the function $a(n)$ is called the *area function* of the machine $\mathcal{S}(M)$.

The union of the S-machines

$$R_{4,\alpha}(\tau), \mathcal{S}_\alpha(\tau), R_{\alpha,\omega}(\tau), \mathcal{S}_\omega(\tau), R_{\omega,9}(\tau)$$

will be denoted by $R_{4,9}(\tau)$. The hardware of $R_{4,9}(\tau)$ is the same as for $\mathcal{S}(M)$. The following lemma is a corollary of Lemmas 4.13 and 4.14. The machine $R_{4,9}$ is defined as a composition of submachines in the same manner as machines $\mathcal{S}_4$ and $\mathcal{S}_9$, so this lemma can be proved in a similar way as Lemmas 4.6, 4.12, hence we omit the proof.

Lemma 4.16 *Let τ be of the form (22). Let W be a positive word to which the rule $R_{4,\alpha}(\tau)$ is applicable. Let W' be obtained from W by first applying $R_{4,\alpha}(\tau)$ and then replacing (τ, α) in all state letters by $(\tau, 9)$. Then*

1. *Every computation of $R_{4,9}(\tau)$ starting with W or W' is semiproper. The length of any such computation C is bounded by $O(|W| + |V|)$ and the area is bounded by $O((|W| + |V|)^2)$ where V is the last word in the computation.*

2. *There is only one reduced computation C of $R_{4,9}(\tau)$ starting with W and such that the rule $R_{\omega,9}(\tau)^{-1}$ is applicable to the last word in this computation. The last word in this computation is W'. This computation contains two transitions such that the rule applied in one of these transitions contains a word Ex as one of its left sides, where $E \in \mathbf{E}(0)$ and $x \in \mathbf{X}(0)$, and the rule applied in the other transition has a word $x'F'$ as one of its left sides, where $x' \in \mathbf{X}(k+1)$, $F' \in \mathbf{F}'(k+1)$.*

3. *Every computation of the machine $R_{4,9}(\tau)$ starting with W and ending with a word to which the rule $R_{4,\alpha}(\tau)$ is applicable, is trivial. Every computation of $R_{4,9}(\tau)$ starting with W' and ending with a word to which $R_{\omega,9}(\tau)^{-1}$ is applicable, is trivial.*

The next proposition is the main statement of this section. In this Proposition, we included all the information about the S-machine $\mathcal{S}(M)$ that we'll use later.

Proposition 4.1 *($\mathcal{S}(M)$ simulates M.) Let $M = \langle X, Y, Q, \Theta, \vec{s}_1, \vec{s}_0\rangle$ be a Turing machine satisfying the conditions of Lemma 3.1. Then its time function, generalized time function and generalized space function are equivalent to each other, and its area function is equivalent to the square of the time function $T(n)$. Let $W_0 = \sigma(c_0)$ where c_0 is the accept configuration of the Turing machine M. Then:*

1. *Every computation of $\mathcal{S}(M)$ starting at W_0 consists of normal words and is semiproper. If a rule $R_{4,\alpha}(\tau)$ is applied in this computation then the word W to which it applies is positive, the degree of α in this word is positive, and the result of the application of this rule is shorter than W.*

2. *A configuration c of the machine M is acceptable by M if and only if $\mathcal{S}(M)$ can take $\sigma(c)$ to W_0.*

3. *If c is an acceptable configuration of M then there exists a one-to-one correspondence ψ between all accepting computations of M starting with c and all computations of $\mathcal{S}(M)$ connecting $\sigma(c)$ and W_0. This correspondence satisfies the following properties:*

   (a) *For every accepting computation C' of the machine M, all words in the computation $\psi(C')$ are positive. In particular, $\psi(C')$ is proper.*

   (b) *If the accepting computation C' has length T and space S then $\psi(C')$ has length between $\epsilon_1 S^3$ and $\epsilon_2 TS^2$ and area between $\epsilon_3 S^4$ and $\epsilon_4 TS^3$ for some positive constants $\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4$.*

4. *The generalized time function of $\mathcal{S}(M)$ is equivalent to $T(n)^3$ and the area function of $\mathcal{S}(M)$ is equivalent to $T(n)^4$. If there exists a computation of $\mathcal{S}(M)$ connecting W and W_0 then the minimal area of such computations is $\leq \epsilon_5 T^4(\epsilon_6 ||W||) + O(|W|^3)$ for some constants ϵ_5 and ϵ_6.*

5. *Let $C = (W_1, ..., W_n)$, $n \geq 3$, be a reduced computation starting with a positive word W_1 such that there exists a computation of $\mathcal{S}(M)$ connecting W_0 and W_1. Suppose a rule $R_{4,\alpha}(\tau)$ is applied in the transition $W_1 \to W_2$ and a rule $R_{4,\alpha}(\tau')$ is applied in the transition $W_{n-1} \to W_n$. Then there exist $1 < i, j < n$ such that the rule applied in the transition $W_i \to W_{i+1}$ contains a word Ex as one of its left sides, where $E \in \mathbf{E}(0)$ and $x \in \mathbf{X}(0)$ and the rule applied in the transition and $W_j \to W_{j+1}$ has a word $x'F'$ as one of its left sides, where $x' \in \mathbf{X}(k+1)$, $F' \in \mathbf{F}'(k+1)$.*

6. *$R_{4,\alpha}(\tau)^{\pm 1}$ are the only rules in $\mathcal{S}(M)$ which change the degrees of α and ω in admissible words. The parts involving α and ω in this rule have the form $xF \to \alpha^{-1}x_1F$ and $E'x' \to E'x'_1\omega^{-1}$ where $x, x_1 \in \mathbf{X}(0)$, $F \in \mathbf{F}(0)$, $E' \in \mathbf{E}'(k+1)$, $x', x'_1 \in \mathbf{X}(k+1)$.*

7. *The rules involving α (resp. ω) are $R_{4,\alpha}(\tau)$ and the rules of the form $x \to \alpha^{\pm 1}x\alpha^{\mp 1}$ where $x \in \mathbf{X}(0)$ (resp. $x' \to \omega^{\pm 1}x'\omega^{\mp 1}$ where $x' \in \mathbf{X}'(k+1)$). For every $x \in \mathbf{X}(0) \cup \mathbf{X}(k+1)$ there exist at most two mutually inverse rules involving x and α or x and ω.*

Proof. Let c be a configuration of the machine M. Suppose that a computation C of the machine $\mathcal{S}(M)$ starts with $\sigma(c)$. Notice that all rules of $\mathcal{S}(M)$ take normal words to normal words. Since $\sigma(c)$ is a normal word, every word in C is normal.

The computation C can be represented in the form (1):

$$C_1, C_2, ..., C_N \tag{24}$$

where each C_i is a computation of one of the machines:

$$\mathcal{S}_4(\tau), \mathcal{S}_9(\tau), R_{4,9}(\tau), R_9(\tau), P(\tau')$$

where τ is any transition of M of the form (22), τ' is any transition of M of the form (23).

The submachine which executes the block C_i will be denoted by $\chi(C_i)$. Now we shall describe the sequence $\chi(C) = (\chi(C_1), \chi(C_2), ..., \chi(C_N))$.

By the definition of the transition machines it is clear that every non-trivial reduced computation of each of these machines contains only two words.

Also from the definition it follows that

a) the types of consecutive blocks C_i and $C_{i\pm 1}$ can be the following:

- $R_4(\tau)$ and $\mathcal{S}_4(\tau)$,
- $\mathcal{S}_4(\tau)$ and $R_{4,9}(\tau)$,
- $R_{4,9}(\tau)$ and $\mathcal{S}_9(\tau)$,
- $\mathcal{S}_9(\tau)$ and $R_9(\tau)$,
- $R_9(\tau)$ and $R_4(\tau_1)$,
- $R_4(\tau)$ and $R_4(\tau_1)$,
- $R_4(\tau)$ and $P(\tau_1)$,
- $R_9(\tau)$ and $R_9(\tau_1)$,
- $R_9(\tau)$ and $P(\tau_1)$.

for some (different) τ and τ_1, and

b) $\chi(C)$ cannot contain subsequences $Z_1Z_2Z_3$ where Z_1, Z_2, Z_3 belong to the set

$$\{R_4(\tau), R_9(\tau), P(\tau) \mid \tau \in \Theta\}.$$

Let us prove the following five statements by induction on i from 1 to N.

A1. if $i < N$ and $\chi(C_i) = \mathcal{S}_4(\tau)$ then the degrees of δ in the (τ, δ)-parts of all words in C_i are > 0 and all words in C_i are positive.

A2. if $i < N$, $\chi(C_i) = \mathcal{S}_9(\tau)$ and C_i starts with a positive word then all words in C_i are positive.

A3. if $i \leq N-2$, or if $i = N-1$ and $\chi(C_i) \neq R_{4,9}(\tau)$, then all words in C_i are positive.

A4. for every i, if $\chi(C_i) = R_{4,9}(\tau)$ then the first word in C_i is positive; moreover the degree of α in the word W from C_i, to which $R_{4,\alpha}(\tau)$ is applicable, is positive, and the result of the application of this rule is shorter than W.

A5. If $1 < i < N$ and $\chi(C_i) \in \{\mathcal{S}_4, \mathcal{S}_9, R_{4,9}\}$ then $\chi(C_i) \neq \chi(C_{i+1})$.

All five statements A1 – A5 hold for $i = 1$ because of Lemmas 4.6, 4.12, 4.16.

Suppose that $i < N$ and $\chi(C_i) = \mathcal{S}_4$. Then

$$\chi(C_{i+1}), \chi(C_{i-1}) \in \{R_4(\tau), R_{4,9}(\tau)\}.$$

If $\chi(C_{i-1}) = \chi(C_{i+1})$ then Lemma 4.6 (ii) applies and by this lemma C_i is trivial, a contradiction. So $\chi(C_{i-1}) \neq \chi(C_{i+1})$. Then all conditions of Lemma 4.6 apply to the (τ, δ)-part of the first and the last words from C_i. Therefore by this lemma the degree of δ in the first and the last words of C_i are > 0, C_i is a proper computation and the degree of δ in the (τ, δ)-part does not change during C_i. Since the work of $\mathcal{S}_4(\tau)$ affects only the (τ, δ)-parts of admissible words, we deduce that all words in C_i are positive. This proves statement A1.

Similarly suppose that $\chi(C_i) = \mathcal{S}_9(\tau)$. Then by Lemma 4.12 (ii), $\chi(C_{i-1}) \neq \chi(C_{i+1})$ and the τ-parts of the words from C_i are positive. This proves A2.

Suppose that $i \leq N-2$ and $\chi(C_i) = R_{4,9}(\tau)$. Then by Lemma 4.16 either the first rule applied in C_i is $R_{4,\alpha}(\tau)$ or the last rule applied in C_i is $R_{4,\alpha}(\tau)^{-1}$ (depending on whether $\chi(C_{i+1})$ is $\mathcal{S}_9(\tau)$ or $\chi(C_{i+1})$ is $\mathcal{S}_4(\tau)$). In the second case all words in C_i are positive by Lemma 4.16.

Remark. Notice that in this case the degree of α in the last word in C_i is positive because $R_{4,\alpha}(\tau)^{-1}$ inserts a new α in the α-part of the word. Also notice that $R_{4,\alpha}(\tau)^{-1}$ applied to a positive word always makes the word longer.

Suppose that the first rule applied in C_i is $R_{4,\alpha}(\tau)$. We have that $i > 1$ since $\chi(C_1) \in \{R_4(\tau), R_9(\tau), P(\tau) \mid \tau \in \Theta\}$. Clearly $\chi(C_{i-1}) = \mathcal{S}_4$. If $\chi(C_{i-1}) = \chi(C_{i+1})$ then by Lemma 4.16, C_i is trivial. Therefore $\chi(C_{i+1}) = \mathcal{S}_9$. Since the statement A1 has been proved already, the degree of δ in the τ-part of the first word of C_i is strictly positive. Therefore the rule $R_{4,\alpha}(\tau)$ does not insert a negative letter in the (τ, δ)-part.

We know that all words in C are normal. Since the first word in C_i is positive and normal, and the degree of δ is greater than 0, the degree of α and the degree of ω in this word are also positive. Therefore the rule $R_{4,\alpha}$ does not insert a negative letter into the α-part and ω-part.

Remark. Notice that in this case we showed that the degree of α in the word to which $R_{4,\alpha}(\tau)$ applies cannot be 0. Also since the degree of δ in the (τ, δ)-part of the first word of C_i is positive, the application of the rule $R_{4,\alpha}(\tau)$ makes the (τ, δ)-part shorter by one letter. It also makes the α-part and the ω-part shorter by one letter. It can make the $(\tau, \bar{Y})$-part longer by at most one letter and it does not touch other parts of the word. Therefore the resulting word is at least 2 letters shorter than the word to which this rule was applied.

By Lemma 4.16 the (τ,δ)-part of the first word in C_{i+1} is positive. If the (τ, M)-part of this word is not positive then it has the form

$$E(i,\tau,9)ua^{-1}x(i,\tau,9)F(i,\tau,9)$$

where u is a positive word (inherited from the first word in C_i) which does not end with a.

But then Lemma 4.12 implies that C_{i+1} cannot end with a word containing a subword stF (with indices), so C_{i+2} does not exist which contradicts the inequality $i \leq N-2$. Therefore the τ-part of the last word in C_i is positive. Since $R_{4,9}(\tau)$ does not affect tape letters in other parts of admissible words, all words in C_i are positive.

Suppose that $i < N$, $\chi(C_i) \neq R_{4,9}(\tau)$ for any τ, and C_i starts with a positive word.

Then $\chi(C_i) \in \{R_4, R_9, \mathcal{S}_4, \mathcal{S}_9\}$. If

$$\chi(C_i) \in \{P(\tau), R_4(\tau), R_9(\tau)\}$$

then clearly all words in C_i are positive. If $\chi(C_i) = \mathcal{S}_4$ or $\chi(C_i) = \mathcal{S}_9$ then A1 and A2 imply that all words in C_i are positive. This proves A3.

Suppose that $\chi(C_i) = R_{4,9}(\tau)$. If $i \leq N-2$ then all words in C_i are positive by A3 and the degree of α in this word is positive by the two remarks in the proof of A3. If $i = N-1$ or $i = N$ then $\chi(C_{i-1}) \neq R_{4,9}(\tau)$ and by A3, all words in C_{i-1} are positive. In particular, the first word of C_i (which is the same as the last word in C_{i-1}) is positive. If the rule $R_{4,\alpha}(\tau)$ is applied at the beginning of C_i then $\chi(C_{i-1}) = \mathcal{S}_4(\tau)$ and the degree of α in the first word of C_i is positive. If $R_{4,\alpha}(\tau)^{-1}$ is applied at the end of C_i then all words in C_i are positive and so the degree of α in the last word of C_i is positive ($R_{4,\alpha}(\tau)^{-1}$ inserts a new α in the alpha part of the word). This proves A4.

Property A5 follows from A3 and Lemmas 4.6, 4.12, 4.16. Indeed these lemmas imply that if $\chi(C_i) \in \{\mathcal{S}_4, \mathcal{S}_9, R_{4,9}\}$ and $\chi(C_{i-1}) = \chi(C_{i+1})$ then C_i is empty.

Now let us prove the statements of the proposition. Notice that statements 6 and 7 can be proved by a simple inspection of all rules of $\mathcal{S}(M)$, so we need to prove statements 1 – 5 only.

The fact that every word in a computation C starting at W_0 is normal has been established before. If a rule $R_{4,\alpha}(\tau)^{\pm 1}$ is applied in C then it is either the first rule or the last rule applied in a block C_i with $\chi(C_i) = R_{4,9}(\tau)$. If this is the first rule then the word to which this rule is applied is positive by property A4. If this is the last rule then this rule is $R_{4,9}(\tau)^{-1}$. In this case the first word in C_i is positive by property A4, and by Lemma 4.16 there exists exactly one computation of $R_{4,9}(\tau)$ starting at the first word of C_i and ending with a word to which $R_{4,9}(\tau)$ is applicable. In this computation all words are positive. Let us prove that C is semiproper.

By statement A3, if $\chi(C_{N-1}) \neq R_{4,9}(\tau)$ then all words in $\bigcup_{i=1}^{N-1} C_i$ are positive. In particular, the first word in C_N is positive. Then Lemmas 4.6, 4.12, 4.16 imply that the computation C is semiproper.

If $\chi(C_{N-1}) = R_{4,9}(\tau)$ then by statement A4 the first word in C_{N-1} is positive. By statement A3, all words in $\bigcup_{i=1}^{N-2} C_i$ are positive. If the rule $R_{4,\alpha}(\tau)^{-1}$ is applied in C_{N-1} then the first word in C_N is positive and as before C is a semiproper computation. If the rule $R_{4,\tau}(\tau)$ is applied in C_{N-1} and inserts a negative letter a^{-1} in the τ-part of the admissible word then $\chi(C_N) = \mathcal{S}_9(\tau)$, and by Lemma 4.12 the computation C_N is semiproper and does not remove the letter a^{-1}. So the computation C is semiproper.

This proves statement 1 of the proposition.

Notice that property A5 and the list of possible pairs $(\chi(C_i), \chi(C_{i+1}))$ imply that $\chi(C)$ can be represented in the form $D_1(\tau_1), D_2(\tau_2), ..., D_\ell(\tau_\ell)$ where for each $i < \ell$, $\chi(D_i(\tau_i))$ is a sequence of one of the following forms:

(S1) $R_4(\tau_i), \mathcal{S}_4(\tau_i), R_{4,9}(\tau_i), \mathcal{S}_9(\tau_i), R_9(\tau_i),$

(S2) $R_9(\tau_i), \mathcal{S}_9(\tau_i), R_{4,9}(\tau_i), \mathcal{S}_4(\tau_i), R_4(\tau_i),$

(S3) $P(\tau_i),$

and $\chi(D_\ell)$ is a prefix of one of these sequences.

Since all words in C are normal, by Lemma 4.15, there exists a sequence of configurations $C' = (c_1, ..., c_\ell)$ such that the first word in $D_i(\tau_i)$ is $\sigma(c_i)$, $i = 1, ..., \ell$. Notice that if we apply one of the rules of the submachines $R_4(\tau), \mathcal{S}_4(\tau), \mathcal{S}_9(\tau)$ and $R_9(\tau)$ to a positive admissible word W then the corresponding configuration $\mu(W)$ of the machine M will not change. If W is positive and the result W' of an application of $R_{4,\alpha}(\tau)$ is also positive then $\mu(W')$ is obtained from $\mu(W)$ by applying the transition τ. Therefore in C' every c_i is obtained from c_{i-1} by applying a transition from Θ. Thus C' is a computation of the Turing machine M.

If c is an acceptable configuration of the machine M then there exists a computation C' of M from c to c_0. Let $\tau_1\tau_2...\tau_\ell$ be the history of this computation. Then the corresponding computation $D_1(\tau_1), D_2(\tau_2), ..., D_\ell(\tau_\ell)$ takes $\sigma(c)$ to $\sigma(c_0)$. Conversely, if there exists a computation C which takes $\sigma(c)$ to $\sigma(c_0)$ then as we proved before, $C = D_1(\tau_1)...D_k(\tau_\ell)$ where each D_i, $i < \ell$ has one of the forms (S1), (S2), (S3) and D_ℓ is a prefix of such a sequence. Since the last word in D_ℓ is $\sigma(c_0)$, D_ℓ cannot be a proper prefix. As we proved before, the word $\tau_1\tau_2...\tau_\ell$ is the history of a computation which takes c to c_0. This proves statement 2 of the proposition.

Let τ be a positive transition from Θ of the form (22) and suppose that a configuration c' is obtained from c by applying τ. Then Lemmas 4.6, 4.12, 4.16 imply that there exists only one computation of $\mathcal{S}(M)$ of the form (S1) which takes $\sigma(c)$ to $\sigma(c')$ and only one computation of the form $(S2)$ which takes $\sigma(c')$ to $\sigma(c)$. If τ has the form (23) and c' is obtained from c by applying τ then $\sigma(c')$ is obtained from $\sigma(c)$ by applying $P(\tau)$. Therefore the correspondence between computations of M which take c to c_0, and computations of $\mathcal{S}(M)$ which take $\sigma(c)$ to $\sigma(c_0)$, is one-to-one. Let us call this correspondence ψ as in the proposition.

The first property of ψ has been established before.

In order to establish the second property, let $C' = (c_1, ..., c_T)$ be an accepting computation of the machine M of length T and space S. Let $C = \psi(C')$. As before $\psi(C)$ can be represented as a sequence of blocks $D_1(\tau_1), D_2(\tau_2), ..., D_T(\tau_T)$. By Lemmas 4.6, 4.12 the length of each block $D_i(\tau_i)$, $i = 1, ..., T$, in this representation is $O(n_i + n_i^2) = O(n_i^2)$ where n_i is the length of the τ_i-part of the first word in $D_i(\tau_i)$, $i = 1, ..., T$. Since all words in the computation C are normal, the numbers n_i do not exceed $O(S)$. Therefore the length of each block D_i does not exceed $O(S^2)$. Thus the length of C does not exceed $O(TS^2)$. Similarly the area of C does not exceed $O(TS^3)$ because the area of each block does not exceed S^3.

Let c' be a configuration in C' with maximal possible length. By definition of the space of a computation, this length must be equal to S.

Then there exists a number i such that the length of the word written on tape i in the configuration c' is at least S/k where k is the number of tapes of the machine M.

Let $\tau_1...\tau_T$ be the history of the computation C'. Since the computation C' is accepting, the length of the word written on tape i in the last configuration of C' is 0 (see Lemma 3.1, statement 4). Since every transition of the machine M can remove at most one letter from the word written on tape i, there exists at least $S/(2k)-1$ transitions $\tau_{j_1},...,\tau_{j_\ell}$ in the history of computation C' such that the length of the word written on the tape i in configuration c_{j_n}, $n=1,...,\ell$, is at least $S/(2k)$, and the transition τ_{j_n} removes a letter from this word. By Lemmas 4.6, 4.12 and 4.16 the length of each of the corresponding blocks $D_{j_n}(\tau_{j_n})$ exceeds $O(S^2/(4k^2))$ and the area exceeds $O(S^3/(8k^3))$. Therefore the total length of the computation $\psi(C')$ exceeds $O(S^3)$ and the area exceeds $O(S^4)$. This gives us the second property of ψ.

Let us prove the fourth statement of the proposition. Let $d(n)$ be the generalized time function of $\mathcal{S}(M)$, and let $a(n)$ be the area function of $\mathcal{S}(M)$. Let $S(n)$ be the generalized space function, and let $T(n)$ be the time function of the machine M. We know that $S(n)$ is equivalent to $T(n)$. Since every accepting computation ends with all tapes empty, $T(n)\geq n$. It is also clear that $S(n)\geq n$. Thus $T(n)^p$ is equivalent to $S(n)^p$ for every natural number $p>0$.

Fix a number $n>0$. Let c be an accepted configuration of the machine M with $|c|\leq n$, such that the smallest space of an accepting computation for c is $S(n)$. Then by the second property of the function ψ, the shortest computation of $\mathcal{S}(M)$ connecting $\sigma(c)$ with W_0 has length $\geq \epsilon_1 S(n)^3$ and the smallest area computation connecting W_0 and $\sigma(c)$ has area $\geq \epsilon_3 S^4(n)$ for some constants $\epsilon_1,\epsilon_3>0$. Since $|\sigma(c)|=4|c|+13k+6$, we have

$$d(4n+13k+6)\geq \epsilon_1 S^3(n),\ a(4n+13k+6)\geq \epsilon_3 S^4(n)$$

Since the function $S(n)$ is equivalent to $T(n)$, we have

$$T(n)^3\preceq d(n),\ T(n)^4\preceq a(n) \tag{25}$$

On the other hand take any admissible word W, $||W||\leq n$ such that there exists a computation of $\mathcal{S}(M)$ connecting W to W_0. Take the shortest computation C_1 and the smallest area computation C_2 which connect W_0 and W. We can represent C_i, $i=1,2$, in the form $\psi(C_i')C_i''$. Notice that the length of the last configuration c_i in C_i' is smaller than $||W||$. Indeed, recall that C_i'' is a composition of computations of $\mathcal{S}_4(\tau)$, $\mathcal{S}_9(\tau)$, $R_4(\tau)$, $R_{4,9}(\tau)$, $R_9(\tau)$ for some transition τ. Therefore either $\mu(W)$ coincides with c_i, or $\mu(W)$ is a configuration of M obtained from c_i by applying a command τ from Θ, or $\mu(M)$ is obtained from c_i by inserting a negative letter in one of the tapes. Thus $|c_i|\leq|\mu(W)|+1$. On the other hand $|\mu(W)|\leq(||W||-13k-6)/6<|W|-1$. So $|c_i|<||W||$.

The computation C_i', $i=1,2$, is an accepting computation for the configuration c_i. Since $|c_i|\leq n$, there exists an accepting computation $\tilde{C}_i$ for c_i of length $T\leq T(n)$. The space of an accepting computation of length T cannot exceed T (because in the last configuration all tapes are empty and every step of a computation can remove at most one cell). By the second part of our proposition, the length of $\psi(\tilde{C}_i)$ does not exceed $\epsilon_2 T^3 \leq \epsilon_2 T(n)^3$ and the area of this computation does not exceed $\epsilon_4 T^4 \leq \epsilon_4 T(n)^4$. for some constants ϵ_2, ϵ_4. By Lemmas 4.6, 4.12, 4.16 the length of C_i'', $i = 1, 2$, does not exceed $O(|W'| + |W|)^2)$ and the area does not exceed $O((|W'| + |W|)^3)$ where $W' = \sigma(c_i)$. Therefore the length of C_i'', $i = 1, 2$, does not exceed $O(|W|^2)$ and the area does not exceed $O(|W|^3)$. The computation $\psi(\tilde{C}_1)C_1''$ takes W_0 to W. So it cannot be shorter than the computation C_1. Therefore the length of C_1 does not exceed $O(T^3(||W||)) + O(|W|^2)$. Similarly, the area of C_2 does not exceed the area of $\psi(\tilde{C}_2)C_2''$. So the area of C_2 does not exceed $O(T(n)^4) + O(|W|^3)$. This proves the second part of statement 4 of the proposition. Since $||W|| \leq |W|$ and $T(n) \geq n$, we deduce

$$d(n) \preceq T(n)^3, \ a(n) \preceq T(n)^4. \tag{26}$$

The inequalities (25) and (26) imply that $d(n)$ is equivalent to $T(n)^3$ and $a(n)$ is equivalent to $T(n)^4$. This completes the proof of statement 4 of the proposition.

Let us prove property 5 of the proposition. Let $C = (W_1, ..., W_n)$, $n \geq 3$, be a reduced computation of $\mathcal{S}(M)$ starting with a positive word W_1 such that there exists a computation $\bar{C}$ of $\mathcal{S}(M)$ connecting W_0 and W_1. Suppose a rule $R_{4,\alpha}(\tau)$ is applied in the transition $W_1 \to W_2$ and a rule $R_{4,\alpha}(\tau')$ is applied in the transition $W_{n-1} \to W_n$. Let $C = \tilde{C}C'$ and $\bar{C} = C''\tilde{C}^{-1}$ for some computations $\tilde{C}$, C', C'' where $\tilde{C}$ is a maximal prefix of the computation of C which cancels when we reduce the computation $\bar{C}C$. Then $C''\tilde{C}$ and $C''C'$ are two reduced computations starting with W_0.

Suppose first that $\tilde{C}$ is not empty.

Then as before we can decompose the computations $C''\tilde{C}^{-1}$ and $C''C'$ into a sequence of blocks (24). Since the rule $R_{4,\alpha}(\tau)$ is applied in the transition $W_1 \to W_2$, the block B of the computation $C''C'$ containing W_1 is such that $\chi(B) = R_{4,9}(\tau)$. Similarly W_{n-1}, W_n are contained in a block B' of the computation $C''\tilde{C}^{-1}$ such that $\chi(B') = R_{4,9}(\tau')$. Since there is only one rule from $R_{4,9}(\tau')$ applicable to W_{n-1}, the block B' has length 2.

If B is contained in $\tilde{C}^{-1}$ then by Lemma 4.16, C' contains two transitions $W_i \to W_{i+1}$ and $W_j \to W_{j+1}$ such that the corresponding rules have Ex and $x'F'$ as their left sides where $E \in \mathbf{E}(0)$ and $x \in \mathbf{X}(0)$, $x' \in \mathbf{X}(k+1)$, $F' \in \mathbf{F}'(k+1)$ as desired.

Suppose that $\tilde{C}^{-1}$ is not empty and is contained in B. Let T_1, T_2 be the first two words of the block B. Let B'' be the block of the computation $C''C'$ containing T_1 and T_2. Since the rule applied in the transition $T_1 \to T_2$ belongs to $R_{4,9}(\tau)$, $\chi(B'') = R_{4,9}(\tau)$. Since the block B' has length 2, the block B'' is not the last block in the computation $C''C'$. By property A3, T_1 is a positive word. Therefore B'' must end with an application of the rule $R_{4,\alpha}(\tau)^{-1}$. By Lemma 4.16, there is only one reduced computation of the machine $R_{4,9}(\tau)$ which starts with T_1 and ends with a word to which $R_{4,\alpha}(\tau)$ is applicable. Thus $B = B''$. But this contradicts the assumption that $\tilde{C}$ is the longest part of C which cancels in the product $\bar{C}C$.

Finally suppose that $\tilde{C}$ is empty, that is, the computation $\bar{C}C$ is reduced. Then the words W_1 and W_2 are contained in a block B of the computation $\bar{C}C$ and $\chi(B) = R_{4,9}(\tau)$. As before this is not the last block in the computation $\bar{C}C$ because W_{n-1} must belong to another block. Therefore we can apply Lemma 4.16 and conclude that B contains two desired transitions.

This completes the proof of the proposition. $\Box$.

5 The Group Presentation

In this section we convert the S-machine $\mathcal{S}(M)$ into a group presentation.

Let $\mathcal{S} = \mathcal{S}(M)$ be the S-machine described in Proposition 4.1. Let Y be the vector of sets of tape letters, and let Q be the vector of sets of state letters of $\mathcal{S}$. The vector Q has $15k+6$ components which we shall denote by $Q_1, ..., Q_{15k+6}$. Notice that $Q_1 = \mathbf{E}(0)$, $Q_2 = \mathbf{X}(0)$, $Q_3 = \mathbf{F}(0)$, $Q_{15k+4} = \mathbf{E}'(k+1)$, $Q_{15k+5} = \mathbf{X}(k+1)$, $Q_{15k+6} = \mathbf{F}'(k+1)$.

Let W_0 be the same admissible word as in Proposition 4.1. We shall not use the whole definition of $\mathcal{S}$, only Proposition 4.1.

Let Θ be the set of rules of $\mathcal{S}$. Let us call one of each pair of mutually inverse rules from Θ *positive* and the other one *negative*. The set of all positive rules will be denoted by Θ_+ and the set of all negative rules will be denoted by Θ_-.

Let N be any positive integer. Let

$$A = \bigcup_{i=1}^{15k+6} Q_i \cup \{\alpha, \omega, \delta\} \cup \bigcup_{i=1}^{k} Y_i \cup \{\kappa_j \mid j = 1, \ldots, 2N\} \cup \Theta_+.$$

Our group $G_N(\mathcal{S})$ is generated by the set A subject to the set $\mathcal{P}_N(\mathcal{S})$ of relations described below.

1. Transition relations. These relations correspond to elements of Θ_+.

Let $\tau \in \Theta_+$, $\tau = [U_1 \to V_1, ..., U_p \to V_p]$. Then we include relations $U_1^\tau = V_1, ..., U_p^\tau = V_p$ into $\mathcal{P}_N(\mathcal{S})$. Here x^y stands for $y^{-1}xy$. If for some j from 1 to $15k+6$ the letters from Q_j do not appear in any of the U_i then also include the relations $q_j^\tau = q_j$ for every $q_j \in Q_j$.

2. Auxiliary relations.

These are all possible relations of the form $\tau x = x\tau$ where $x \in \{\alpha, \omega, \delta\} \cup \bigcup_{i=1}^k Y_i$, $\tau \in \Theta_+$ and all relations of the form $\tau\kappa_i = \kappa_i\tau$, $i = 1, ..., 2N$, $\tau \in \Theta_+$.

3. The hub relation.

For every word u let $K(u)$ denote the following word:

$$K(u) \equiv (u^{-1}\kappa_1 u\kappa_2 u^{-1}\kappa_3 u\kappa_4 \ldots u^{-1}\kappa_{2N-1}u\kappa_{2N})(\kappa_{2N}u^{-1}\kappa_{2N-1}u \ldots \kappa_2 u^{-1}\kappa_1 u)^{-1}. \quad (27)$$

Then the hub relation is

$$K(W_0) = 1.$$

Figure 1 shows the diagram corresponding to the hub relation if $N = 1$.

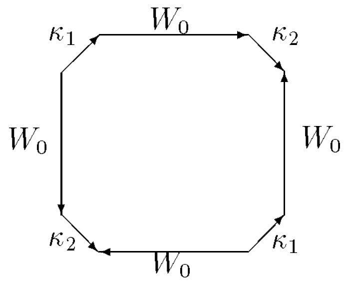

Fig. 1.

We define the presentation $\mathcal{P}_N(\mathcal{S})$ by taking the set of all the relations defined above, their cyclic shifts and inverses of these cyclic shifts. We shall say that a relation is *determined* by some characteristic (a subword, a pair of letters, etc.) if it is determined up to taking cyclic shifts and inverses. Cyclic shifts and inverses of transition (resp. auxiliary and hub) relations will also be called *transition* (resp. *auxiliary* and *hub*) relations.

The group $G_N(\mathcal{S})$ with which we will be working in this paper is given by the presentation $\mathcal{P}_N(\mathcal{S})$.

The following lemma contains some properties of $\mathcal{P}_N(\mathcal{S})$ which either follow from the definition of S-machines or are easy to verify.

Lemma 5.1 *a) Every relation except the hub contains exactly two $\Theta^{\pm 1}$-letters: a letter from Θ and a letter from Θ^{-1}.*

b) If transition relations $s_1, s_2 \in \mathcal{P}_N(\mathcal{S})$ have a common Θ-letter and a common Q_j-letter then s_1 is a cyclic shift of $s_2^{\pm 1}$.

c) If two auxiliary relations $s_1, s_2 \in \mathcal{P}_N(\mathcal{S})$ have a common 2-letter prefix then $s_1 = s_2$.

d) If s_1 and s_2^{-1} are two cyclic shifts of the hub or its inverse and have a common prefix of the form $\kappa_j W_0 \kappa_s$ then $s_1 = s_2$.

6 Some Basic Definitions

We mainly use notation and definitions from Ol'shanskii [23]. In particular, we use the so called *0-refinement* of a van Kampen diagram. We do not want to define it here precisely recall only that a 0-edge is an edge labelled by 1, and a zero-cell corresponds to a relation of the form $a1^n a^{-1} 1^m$ where a is one of the generators, m and n are integers. One can insert zero-edges and zero-cells in a diagram in order to separate two paths which touch each other, or in order to make a path, which touches itself, simple. When we compute an area or diameter of a diagram we never take zero-cells and zero-edges into account.

If Δ is an (ordinary) van Kampen diagram then $\partial(\Delta)$ denotes its boundary. If Δ is an annular diagram then $\partial_o(\Delta)$ and $\partial_i(\Delta)$ denote the outer and the inner boundaries of Δ. The cells of diagrams over $\mathcal{P}_N(\mathcal{S})$ will be called by the names of the corresponding relations: transition cells, auxiliary cells, hub cells (or simply hubs).

We always assume that boundaries of van Kampen diagrams and boundaries of cells are oriented clockwise (this is an insignificant difference with [23]). The *contour* of a cell or a diagram is the union of the boundary and and its inverse.

A path in a diagram Δ is called *simple* if it does not cross itself. A path is called *reduced* if it does not contain consecutive mutually inverse edges. Every path in a diagram can be reduced by removing subpaths of the form ee^{-1}. It is mentioned in [23] that by a 0-refinement one can turn every simple path into an *absolutely simple* path which does not cross and does not touch itself.

The length of a path p, denoted by $|p|$ is the number of non-zero edges in it.

We shall use the following operations which can be performed on arbitrary diagrams.

Taking the inverse (mirror image). Let Δ be a van Kampen diagram over a symmetric set of defining relations $\mathcal{P}$. Consider the mirror image Δ^{-1} of the graph Δ with respect to some straight line. Since $\mathcal{P}$ is symmetric, the graph Δ^{-1} is again a van Kampen diagram over $\mathcal{P}$. We call Δ^{-1} the *inverse of* Δ. It is easy to see that every two inverses of the same diagram Δ can be transformed into each other by a homotopy of a plane, so they are *homotopic*.

Composition. Let Δ_1 and Δ_2 be van Kampen diagrams over a presentation $\mathcal{P}$. Let $\partial(\Delta_1) = p_1p_1'$, $\partial(\Delta_2) = p_2^{-1}p_2'$. Suppose that $\mathrm{Lab}(p_1) = \mathrm{Lab}(p_2)$ in the free group. Here $\mathrm{Lab}(p)$ is the label of the path p. Then by a zero refinement we can make both paths p_1 and p_2 absolutely simple and the labels of p_1 and p_2 identical. After that we can glue Δ_1 and Δ_2 by identifying the corresponding edges of p_1 and p_2. This operation will be called the *composition* and the resulting diagram will be denoted by $\Delta_1 \circ_{p_1=p_2} \Delta_2$.

Let Δ be any van Kampen diagram. The following definition is similar to the definition of a dual graph of a diagram [20], [23]. Fix a point in each of the cells in Δ and a point in the inside of each of the edges of Δ. For each cell π and each edge e on the boundary of this cell, fix a simple polygonal line $\ell(\pi, e)$ inside π_i which connects the point inside π_i and the point inside e. We can choose these lines in such a way that $\ell(\pi, e)$ and $\ell(\pi, e')$ do not have common points except for the fixed point inside π.

The next definition of a band in a diagram is crucial for our paper.

Let S be a set of letters and let Δ be a van Kampen diagram. Fix pairs of S-edges in some cells from Δ (we assume that each of these cells contains at least two S-edges).

Suppose that Δ contains a sequence of cells $(\pi_1, \ldots, \pi_n)$ such that for each $i = 1, \ldots, n-1$ the cells π_i and π_{i+1} have a common S-edge and this edge belongs to the pair of S-edges fixed in π_i and π_{i+1}. Consider the line which is the union of the lines $\ell(\pi_i, e_i)$ and $\ell(\pi_{i+1}, e_i)$, $i = 1, \ldots, n-1$. This polygonal line is called the *median* of this sequence of cells. Then our sequence of cells $(\pi_1, \ldots, \pi_n)$ with common edges $e_1, \ldots, e_{n-1}$ is called an *S-band* if the median is a simple curve or a simple closed curve.

We say that two bands *cross* if their medians cross. We say that two bands *touch* each other if their medians touch each other. We say that a band is an *annulus* if its median is a closed curve.

Let $\mathcal{B}$ be an S-band with common edges $e_1, e_2, \ldots, e_n$ which is not an annulus. Then the first cell has an S-edge e which forms a pair with e_1 and the last cell of $\mathcal{B}$ has an edge f which forms a pair with e_n. Then we shall say that e is the *start edge* of $\mathcal{B}$ and f is the *end edge of* $\mathcal{B}$. If p is a path in Δ then we shall say that a band starts (ends) on the path p if e (resp. f) belongs to p.

If S and T are two disjoint sets of letters, $(\pi, \pi_1, \ldots, \pi_n, \pi')$ is an S-band and $(\pi, \gamma_1,$

$\dots,\ \gamma_m,\ \pi')$ is a T-band then we say that these two bands form an (S,T)-*annulus* if the medians of these bands form a simple closed curve called the *median of the* (S,T)-*annulus*; and in addition the start and end edges of these bands are not contained in the polygon bounded by this median.

If ℓ is the median of an S-annulus or an (S,T)-annulus then the maximal subdiagram of Δ contained in the area bounded by ℓ is called the *inside diagram* of the annulus.

The union of cells of $\mathcal{B}$ forms a subdiagram (perhaps after some 0-refinement). The reduced boundary of this diagram, which we shall call the *boundary of the band*, has the form $e^{\pm 1}pf^{\pm 1}q^{-1}$ (recall that we trace boundaries of diagrams clockwise). Then we say that p is the *top path* of $\mathcal{B}$, denoted by $\mathbf{top}(\mathcal{B})$, and q is the *bottom path* of $\mathcal{B}$, denoted by $\mathbf{bot}(\mathcal{B})$.

Suppose that an S-band $\mathcal{B}$ is a union of two S-bands $\mathcal{B}_1$ and $\mathcal{B}_2$ and the end edge of $\mathcal{B}_1$ coincides with the start edge of $\mathcal{B}_2$. Then the top (bottom) path of $\mathcal{B}$ is obtained by multiplying the top (bottom) paths of $\mathcal{B}_1$ and $\mathcal{B}_2$ and reducing consecutive mutually inverse edges.

If $\mathcal{B}=(\pi_1,\dots,\pi_n)$ is an S-band then $(\pi_n,\pi_{n-1},\dots,\pi_1)$ is also an S-band which we shall call the *inverse of* $\mathcal{B}$. We shall denote this band by $\mathcal{B}^{-1}$.

We shall call an S-band *maximal* if it is not contained in any other S-band. If an S-band $\mathcal{W}$ starts on the contour of a cell π, does not contain π and is not contained in any other S-band with these properties then we call $\mathcal{W}$ a *maximal S-band starting on the contour of* π.

Now let us return to our presentation $\mathcal{P}_N(\mathcal{S})$. Consider the following partition of the generating set A:

$$\Theta \cup Q_1 \cup \ldots \cup Q_{15k+6} \cup \bigcup Y_i \cup \{\kappa_1\} \ldots \cup \{\kappa_{2N}\}. \tag{28}$$

Notice that these subsets are disjoint (the fact that $Q_1,\ldots,Q_{15k+6}$ are disjoint follows from the definition of S-machines).

Let B be a block of this partition. In order to consider B-bands and annuli, we need to divide B-letters of some relation of $\mathcal{P}_N(\mathcal{S})$ into pairs.

Every relation except for the hub contains a letter from Θ and the inverse of this letter. These letters form a Θ-pair.

Let us denote $\{\alpha,\omega,\delta\}\cup\bigcup Y_i$ by $\bar{Y}$. Every auxiliary relation containing a $\bar{Y}$-letter, contains a $\bar{Y}$-letter and the inverse of this letter. These letters form a $\bar{Y}$-pair.

Every auxiliary relation containing κ_i contains also κ_i^{-1}. These two letters form a κ_i-pair.

If a transition relation has a letter from Q_j then it has exactly two letters from $Q_j^{\pm 1}$ (this follows from the definition of an S-rule). They form a Q_j-pair.

The hub relation contains an occurrence of κ_i and an occurrence of κ_i^{-1}. These occurrences form a κ_i-pair in the hub.

A κ_j-band is called *even* (resp. *odd*) if j is even (resp. odd). The number j will be called the *index* of this κ-band.

7 Forbidden Bands and Annuli

In this section we shall consider van Kampen diagrams with a fixed boundary label over the presentation $\mathcal{P}_N(\mathcal{S})$. Let Δ be such a diagram.

For every $j = 1, \ldots, 2N$ we define an automorphism of the group $G'_N(\mathcal{S})$ given by the presentation $\mathcal{P}_N(\mathcal{S})$ without the hub. Take a perfect $4N$-gon with boundary labelled by the word $K(1) = (\kappa_1...\kappa_{2N})(\kappa_{2N}...\kappa_1)^{-1}$ (see formula (27) for the definition of $K(u)$). Consider the reflection with respect to the axis passing through the mid points of the opposite edges labelled by $\kappa_j^{\pm 1}$. Then this reflection induces a permutation ϕ_j on the set

$$\{\kappa_1, \ldots, \kappa_{2N}\} \cup \{\kappa_1^{-1}, ..., \kappa_{2N}^{-1}\}$$

which satisfies the property $\phi_j(\kappa_i^{-1}) = \phi_j(\kappa_i)^{-1}$. We extend ϕ_j to a permutation of the set $A \cup A^{-1}$ by fixing all other letters from $A \cup A^{-1}$. It is easy to see that ϕ_j takes every relation from $\mathcal{P}_N(\mathcal{S})$, except for the hub, to a non-hub relation from $\mathcal{P}_N(\mathcal{S})$. Therefore these maps induce automorphisms of $G'_N(\mathcal{S})$ which will be also denoted by ϕ_j. One notices also that these automorphisms are involutions (each one is its own inverse).

Lemma 7.1 *If Δ contains a κ_j-annulus then there exists a van Kampen diagram with the same boundary label as Δ, smaller area and the same or smaller diameter.*

Proof. Consider each κ_j-cell as a κ_j-band. Then it is easy to verify (see the list of relations in the presentation) that the label of the top and the label of the bottom paths of this band are images of each other under the automorphism ϕ_j of the free group. Therefore the boundaries of the annular diagram formed by any κ_j-annulus $\mathcal{K}$ have labels one of which is the image of another one under ϕ_j in the free group (when we formed the boundaries of the annular diagram, we reduce pairs of consecutive mutually inverse edges and this does not change the values of the labels of these boundaries in the free group). We can assume that $\mathcal{K}$ is an innermost κ_j-annulus, that is, there are no k_j-annuli inside the subdiagram Δ' bounded by the median of $\mathcal{K}$. Since the contour of $\mathcal{K}$ contains no κ_j-edges, the subdiagram Δ' does not contain κ_j-edges (otherwise the maximal $\mathcal{K}_j$-band containing this edge would have to be an annulus). This implies that Δ' does not contain hubs. That is, Δ' is a van Kampen diagram over the presentation of $G'_N(\mathcal{S})$ (see Figure 2).

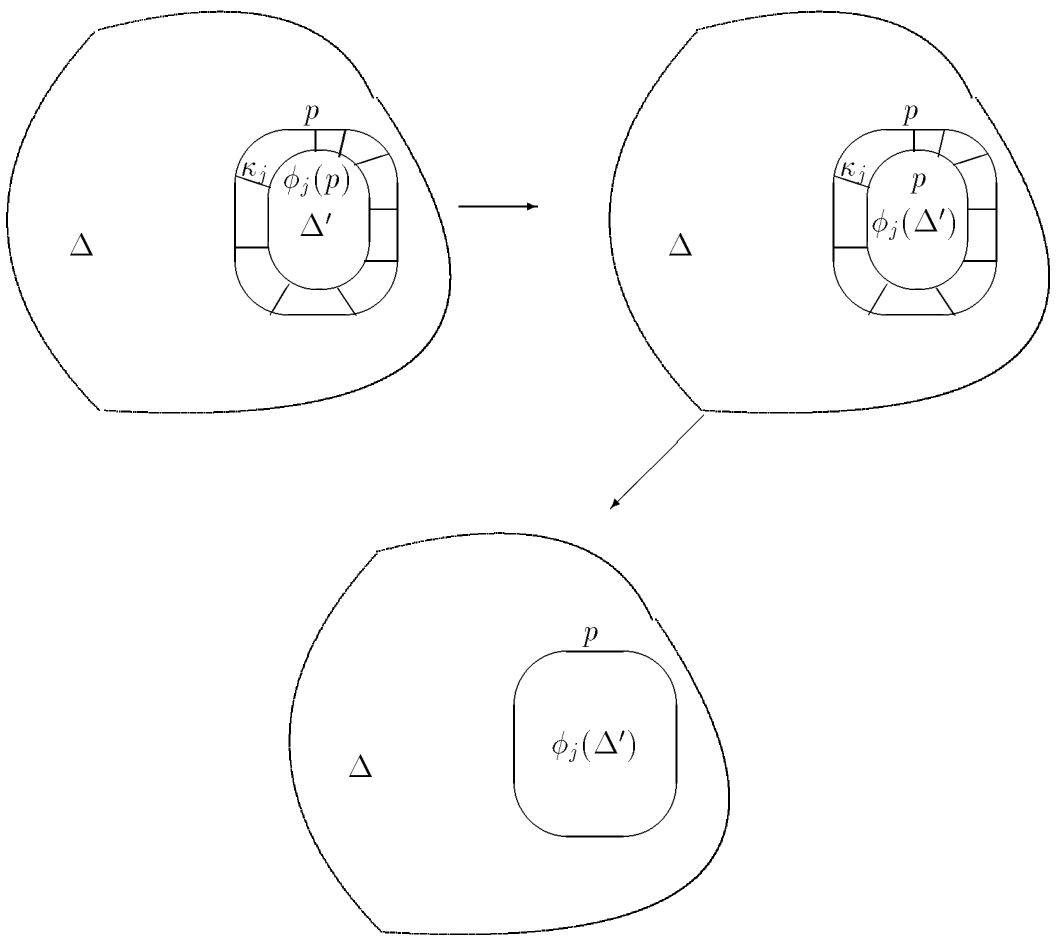

Fig. 2.

Let us apply ϕ_j to all labels of Δ'. Since ϕ_j takes relations of $G'_N(\mathcal{S})$ to relations of this group, the subdiagram Δ' turns into another van Kampen diagram $\phi_j(\Delta')$ over $\mathcal{P}_N(\mathcal{S})$. We can also notice that the labels of the inner and the outer contours of our annulus are now the same in the free group. By applying a 0-refinement one can make these labels graphically equal. After that one can identify these contours and remove all the cells of the annulus from the diagram. The result will be a van Kampen diagram Σ with the same boundary label as Δ and smaller area.

It is also easy to show that the diameter of the diagram can only increase when we remove a κ-annulus. Indeed, with every path p in Δ one can associate a path p' in Σ conecting the same vertices. This path is obtained by removing some edges (the common edges of the removed annulus) and identifying the end vertices of these edges, and by relabeling the parts of p which are contained in the subdiagram bounded by the annulus. The length of p' can be only smaller than the length of p. Therefore the diameter of Σ can be only smaller than the diameter of Δ. $\square$

Now we can define *reduced* van Kampen diagram over $\mathcal{P}_N(\mathcal{S})$.

Definition 7.1 A van Kampen diagram over a presentation $\mathcal{P}$ is *reduced* if it does not contain any of the following:

1. A 0-edge (that is an edge labelled by 1).
2. A *reducible* pair of cells (see Figure 3):

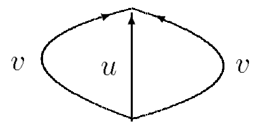

Fig. 3.

Here v is not empty.

3. A κ_i-annulus (for any i).

Lemma 7.1 shows that if Δ has a κ-annulus then its area can be decreased without increasing the diameter. It is easy to see that if Δ contains a reducible pair of cells, its area can decrease too: both cells can be removed by removing the path labelled by u and identifying the paths labelled by v. Unfortunately the diameter of the diagram can increase. This does not cause any difficulty in computing an upper bound for the smallest isodiametric function, but this makes it more difficult to find a good lower bound (see Section 12).

For each word w which is equal to 1 in our group $G_N(\mathcal{S})$ there exists a reduced diagram with boundary label w. Indeed, it is enough to take the van Kampen diagram with minimal number of cells and boundary label w (such a diagram exists by van Kampen's lemma).

Definition 7.2 An annular van Kampen diagram Ψ over $\mathcal{P}_N(\mathcal{S})$ will be called *reduced* if any ordinary van Kampen diagram obtained by cutting Ψ along a simple path connecting $\partial_0(\Psi)$ with $\partial_i(\Psi)$ is reduced.

Notice that if we obtain an annular diagram Ψ by removing a subdiagram from a reduced van Kampen Diagram Δ then Ψ is reduced.

For the rest of this section we shall assume that Δ is a *reduced diagram* over $\mathcal{P}_N(\mathcal{S})$.

Lemma 7.2 *If Δ is reduced and does not have hubs then it does not have Q_j-annuli for any $j = 1, ..., 15k + 6$.*

Proof. Suppose there is a Q_j-annulus. Let $\mathcal{Q}$ be the Q_i-annulus with the smallest inside diagram Δ'. Since Q-annuli cannot cross (this follows from the definition of an S-rule), Δ' does not contain Q-cells (otherwise we would get a Q-annulus with smaller inside diagram). Therefore all cells in Δ' are auxiliary. Since every cell in $\mathcal{Q}$ has exactly two Q_i-edges, the boundary $\partial(\Delta')$ contains no Q-edges. Thus Δ' is a diagram over the presentation consisting of all auxiliary (commutativity) relations. The group H given by this presentation has a homomorphism onto the direct product of the free group on Θ and the free group on $\Gamma \cup \{\alpha, \omega, \kappa_1, \ldots, \kappa_{2N}\}$. Therefore the word $u = \mathrm{Lab}(\partial(\Delta'))$ is equal to 1 in this direct product. Every Q-cell in $\mathcal{Q}$, considered as a Q-band, contains exactly one Θ-edge on the top path (resp. on the bottom path). If two of these Θ-edges cancel when we form the top/bottom path of $\mathcal{Q}$, then the corresponding cells of $\mathcal{Q}$ cancel, by Lemma 5.1 (b). This cannot happen since Δ is reduced. Therefore u contains a Θ-letter.

This implies that u contains a subword of the form $\tau^{\pm 1} v \tau^{\mp 1}$ for some letter $\tau \in \Theta_+$ and some word v which does not contain Θ-letters. Then the Θ-letters in this subword must label edges of consecutive cells π and π' in $\mathcal{Q}$ (again we use the fact that Θ-edges do not cancel when we form the top/bottom path of $\mathcal{Q}$). Then the cells π and π' cancel. Indeed by Lemma 5.1 (b) the relation s_1 corresponding to the cell π_1 is a cyclic shift of $s_2^{\pm 1}$ where s_2 is the relation corresponding to π_2. We can assume that we start reading s_1 and s_2 at the beginning of their common Q_j-edge. Since the Θ_+-edges on the bottoms of these cells are oriented toward each other, we conclude that $s_1 = s_2^{-1}$ and each of the words s_1 and s_2 contains only two Θ-edges, $s_1 = s_2^{-1}$. So the cells π_1 and π_2 cancel.

This again contradicts the assumption that Δ is reduced. $\Box$

Lemma 7.3 *If Δ is reduced, does not have hubs and $\partial(\Delta)$ consists of Θ-edges then Δ does not have cells.*

Proof. Indeed, by Lemmas 7.2 and 7.1, Δ has no Q_j-annuli. Thus it cannot contain Q_j-cells. Therefore all cells in Δ are auxiliary. Hence, as in the proof of Lemma 7.2, the boundary label of Δ is equal to 1 in the free group generated by Θ. Since Δ is reduced, it does not contain cells. $\Box$

Lemma 7.4 *If Δ is reduced and does not contain hubs, then it does not have (Q_j, Θ)-annuli.*

Proof. Suppose that Δ contains a (Q_j, Θ)-annulus $\mathcal{W}_1 \cup \mathcal{W}_2$ where $\mathcal{W}_1$ is a Q_j-band and $\mathcal{W}_2$ is a Θ-band. Let Δ' be the inside diagram of this annulus. We can assume that Δ' has the smallest possible area.

Then the contour of Δ' does not contain Q-edges (otherwise there would be a (Q_i, R)-annulus with a smaller inside diagram). Therefore Δ' does not contain any transition cells (by Lemma 7.2 there are no Q-annuli). So it can only contain auxiliary cells. Thus Δ' is a diagram over the group H given by the auxiliary relations only. We know from the proof of Lemma 7.2 that this group has a homomorphism onto the direct product of the free group over Θ and the free group generated by other letters (not from $\bigcup Q_i \cup \Theta$). Thus if we remove all non-Θ-letters from the label w' of the boundary of Δ', we get a word which is equal to 1 in the free group generated by Θ. Therefore w' contains a subword $t = \tau^{\pm 1} v \tau^{\mp 1}$ as in Lemma 7.2 or else the contour of Δ' does not have Θ-edges at all.

Suppose first that the contour of Δ' contains Θ-edges. The part of the contour of Δ' which is contained in the contour of $\mathcal{W}_2$ cannot contain Θ-edges because the contour of each Θ-cell contains exactly two Θ-edges which form an opposing pair. Thus both τ's in the subword t must be labels of some edges on the part of $\partial(\Delta')$ which is contained in $\mathcal{W}_1$. These two τ's must belong to two neighboring transition cells, so Δ is not reduced (we apply the same argument as in Lemma 7.2).

Now suppose that $\partial(\Delta')$ does not contain Θ-edges (see Figure 4).

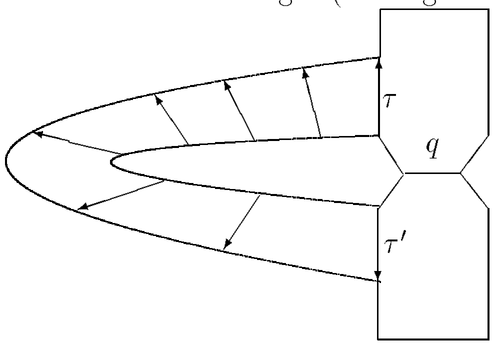

Fig. 4.

Then $\mathcal{W}_1$ contains just 2 cells, the intersection cells of $\mathcal{W}_1$ and $\mathcal{W}_2$. Then the edges labelled by τ and τ' on Fig. 4 belong to $\mathcal{W}_2$. The τ's in every Θ-cell of $\mathcal{W}_2$ are the same. Therefore $\tau = \tau'$. This implies that the two Q-cells in $\mathcal{W}_1$ form a reducible pair (Lemma 5.1 b)), a contradiction. □

Lemma 7.5 *Suppose that (not necessarily reduced) Δ contains a (κ_j, Θ)-annulus $\mathcal{A}$ consisting of a κ_j-band $\mathcal{W}_1$ and a Θ-band $\mathcal{W}_2$. Suppose that $\mathcal{W}_1$ does not have hubs. Then $\mathcal{W}_1$ contains a reducible pair of cells which belong to the same Θ-band.*

Proof. If there is a κ_j-cell in $\mathcal{W}_1$ between the intersection cells ρ_1 and ρ_2 of $\mathcal{W}_1$ and $\mathcal{W}_2$ then this cell will belong to a Θ-band which forms a (κ_j, Θ)-band with a part of $\mathcal{W}_1$ and the inside diagram of this annulus would be smaller than the inside diagram of $\mathcal{A}$. Therefore we can assume that the cells ρ_1 and ρ_2 are consecutive cells on $\mathcal{W}_1$. The start and the end edges of $\mathcal{W}_2$ form an opposing pair with the same label. Therefore the cells ρ_1 and ρ_2 correspond to mutually inverse relations and have a common κ_j-edge. Thus these cells form a reducible pair and belong to the same Θ-band. □

Lemma 7.6 *If Δ is reduced and does not contain hubs then it does not contain Θ-annuli.*

Proof. Take an innermost Θ-annulus $\mathcal{B}$. Let Δ' be the inside subdiagram formed by this annulus. Then Δ' does not contain Θ-annuli. Since the contour of Δ' contains no Θ-edges, the diagram Δ' contains no Θ-edges either. Since the only cells that contain no Θ-edges are the hubs (which were ruled out here), Δ' contains no cells. Therefore $\mathrm{Lab}(\partial(\Delta')) = 1$ in the free group. Since Δ contains no (Q, Θ)-annuli (Lemma 7.4), $\partial(\Delta')$ contains no Q-edges. Therefore $\partial(\Delta)$ contains only $(\bar{Y})$-edges and all cells of $\mathcal{B}$ are auxiliary. Since $\mathrm{Lab}(\partial(\Delta')) = 1$ there are two neighboring edges in $\partial(\Delta')$ with mutually inverse edges. The cells of $\mathcal{B}$ containing these two edges form a reducible pair, by Lemma 5.1 (c). This contradicts the assumption that Δ is reduced. □

Lemma 7.7 *If Δ is reduced, does not contain hubs and $\partial(\Delta)$ has no Θ-edges then Δ does not have cells.*

Proof. Since Δ contains no Θ-annuli (Lemma 7.6), the assumptions of the Lemma imply that Δ contains no Θ-edges at all. Since every relation in $\mathcal{P}_N(\mathcal{S})$ except for the hub contains letters from Θ, the diagram Δ has no cells. $\square$

Lemma 7.8 *If Δ is reduced and contains no Q-edges then it does not have $(\bar{Y}, \Theta)$-annuli.*

Proof. Since Δ contains no Q-edges, all cells in Δ are auxiliary. Take an innermost $(\bar{Y}, \Theta)$-annulus. Let Δ' be the inside diagram of this annulus. Then $\partial(\Delta') = pq$ where $\mathrm{Lab}(p)$ contains no Θ-edges and $\mathrm{Lab}(q)$ contains no $\bar{Y}$-edges (p is the bottom or the top path of a Θ-band, q is the bottom or the top path of a $\bar{Y}$-band). It follows from the structure of auxiliary $\bar{Y}$-cells that the top and the bottom paths of every non-empty $\bar{Y}$-band must contain Θ-edges. Therefore either q contains Θ-edges or q is empty. The first case is impossible because our $(\bar{Y}, \Theta)$-annulus was innermost. Therefore q is empty. Then $\partial(\Delta')$ contains no Θ-edges. By Lemma 7.7, Δ' contains no cells. Therefore $\mathrm{Lab}(\partial(\Delta')) = 1$ in the free group. This implies, as before, that our $(\bar{Y}, \Theta)$-annulus is not reduced. $\square$

Lemma 7.9 *If Δ is reduced and does not have hubs then it does not have $\bar{Y}$-annuli.*

Proof. Suppose that Δ has a $\bar{Y}$-annulus. Take the innermost $\bar{Y}$-annulus $\mathcal{Y}$. Then the outer contour p of this annulus cannot contain Q-edges since a $\bar{Y}$-band cannot contain transition cells (by the definition of $\bar{Y}$-bands). Lemma 7.2 implies that the subdiagram bounded by p does not contain Q-edges. It also cannot contain κ- or Θ-edges because of Lemmas 7.8 and 7.6. Thus the diagram bounded by p must be empty, a contradiction (this diagram contains $\mathcal{Y}$). $\square$

8 Diagrams Without Hubs

In this section, we estimate the area and diameter of a diagram without hubs over the presentation $\mathcal{P}_N(\mathcal{S})$ and also consider the process of reducing a non-reduced diagram without hubs.

With every diagram Δ over $\mathcal{P}_N(\mathcal{S})$ we associate two numbers: $n(\Delta)$ is the length of the boundary of Δ and $h(\Delta)$ is the number of hubs in Δ.

Lemma 8.1 *Let Δ be a reduced diagram without hubs over the presentation $\mathcal{P}_N(\mathcal{S})$, with perimeter $n = n(\Delta)$ and with m maximal Q-bands. Then $m \leq n$ and the area of Δ is at most Cn^2m where C is a constant. Therefore the area of Δ does not exceed Cn^3. The diameter of Δ does not exceed $C'n$ for some constant C'.*

Proof. Every cell in Δ belongs to a maximal Θ-band. Since Δ contains no Θ-annuli (Lemma 7.6), each of these Θ-bands starts and ends on the contour of Δ. Therefore there are at most n maximal Θ-bands. A similar reasoning shows that $m \leq n$.

Every transition cell is an intersection of a Θ-band and a Q_j-band. Since Δ does not contain (Q_j, Θ)-annuli (Lemma 7.4), each Θ-band intersects each Q_j-band at most once. Therefore Δ contains at most $mn/4$ transition cells (the total number of intersections is bounded by mn but each transition cell is counted at least 4 times because it belongs to two mutually inverse Q_j-bands and to two mutually inverse Θ-bands).

Let Φ be the set of all maximal $\bar{Y}$-bands. Every non-transition cell with $\bar{Y}$-edges in Δ belongs to two mutually inverse bands in Φ. Therefore we need to compute the number of bands in Φ and their lengths.

Since Δ does not contain $\bar{Y}$-annuli (Lemma 7.9), each of the $\bar{Y}$-bands in Φ starts either on the contour of Δ or on the contour of one of the transition cells. The number of $\bar{Y}$-edges on a transition cell is bounded by a constant. Therefore the number of $\bar{Y}$-bands in Φ is at most Cmn for some constant C. Every cell in each of the bands in Φ is an intersection of a $\bar{Y}$-band and a Θ-band. By Lemma 7.8, each of the $\bar{Y}$-bands in Φ intersects each of the Θ-bands in Δ at most once (we use the fact that $\bar{Y}$-bands in Φ do not contain transition cells, and Lemma 7.4). So the length of each of the bands in Φ does not exceed $n/2$. Therefore the number of non-transition $\bar{Y}$-cells in Δ does not exceed C_1mn^2 for some constant C_1.

Each non-transition cell which does not contain $\bar{Y}$-edges is the intersection of a Θ-band and a κ_j-band. Since Δ does not contain κ-annuli, the total number of maximal κ-bands (including inverses) is at most n. By Lemmas 7.5 each Θ-band intersects each of the κ-band at most once. Therefore the number of non-transition cells without $\bar{Y}$-edges in Δ does not exceed $n^2/4$.

Thus the total number of cells in Δ does not exceed $C_1mn^2 + n^2/4 < C_2mn^2$.

In order to estimate the diameter of Δ pick a vertex v in Δ. Let d be the distance from v to the boundary of Δ. It is clear that $n/2$ plus the maximal among these d's for all vertices v is the upper bound for the diameter of Δ, so we need to estimate d. Without loss of generality we can assume that v is not on the boundary of Δ. The vertex v belongs either to a κ-cell, or to a $\bar{Y}$-cell. In the first case v belongs to the boundary of a κ-band in Δ. Since the length of this κ-band is at most n, we conclude that $d \leq n$.

In the second case v is within a constant distance d_1 from a $\bar{Y}$-edge e. Consider the maximal $\bar{Y}$-band $\mathcal{A}$ containing e. We know that the length d_2 of this band is at most n. The length of the boundary of this band is $O(n)$. The band $\mathcal{A}$ ends either on the contour of the diagram Δ or on the boundary of a transition cell. In the first of these cases $d \leq d_1 + O(n) \leq O(n)$. In the second case we can conclude that v is within distance $O(n)$ from the boundary of a Q-band. As we know, the length of every maximal Q-band in Δ is at most n. Therefore the length of the boundary of any Q-band is $\leq O(n)$. Since every maximal Q-band in Δ ends and starts on the contour of Δ, we can conclude that $d \leq O(n)$. Therefore the diameter of Δ is $O(n)$.

The lemma is proved. $\square$

Later we shall need to analyze the process of reducing a non-reduced diagram without hubs over $\mathcal{P}_N(\mathcal{S})$. First we prove the following general results.

Lemma 8.2 *Let Ψ be a non-reduced diagram over any presentation. Consider a process of cancelling a sequence of reducible pairs of cells in Ψ. If two edges e and f in Ψ*

get identified in this process then there exists a path t in Ψ connecting the initial vertices $\imath(e)$ and $\imath(f)$ and such that $\mathrm{Lab}(t)=1$ *in the free group.*

Proof. We shall use induction on the length of the cancellation process. If $e=f$ in Ψ (that is if the length of the process is 0) then the statement is obvious. Suppose that $e\neq f$ in Ψ and e is identified with f after c cancellations. Let Ψ' be the diagram obtained from Ψ after $c-1$ cancellations. There exist a pair of cells π and π' in Ψ' such that $e^{-1}\in\partial(\pi)$, $f\in\partial(\pi')$, π and π' form a reducible pair in Ψ' and e and f are identified after we cancel π and π'. Since π and π' form a reducible pair in Ψ', the boundaries of these cells have a common edge g. Since e and f are corresponding edges of π and π' the path which connects $\imath(g)$ with $\imath(e)$ on $\partial(\pi)^{-1}$ (oriented counterclockwise) and the path connecting $\imath(g)$ with $\imath(f)$ on $\partial(\pi')$ have equal labels. Let π_1 and π_1' be the "preimages" of the cells π and π' in Ψ. Then these cells contain edges g_1 and g_2 respectively such that

- the path t connecting $\imath(g_1)$ with $\imath(e)$ on $\partial(\pi_1)^{-1}$ and the path t' connecting $\imath(g_2)$ with $\imath(f)$ on $\partial(\pi_1')$ have equal labels and
- the edges g_1 and g_2 are identified (with the edge g) after $c-1$ cancellations.

By the induction hypothesis there exists a path p in Ψ connecting $\imath(g_1)$ with $\imath(g_2)$ whose label is equal to 1 in the free group. Then the path $t^{-1}pt'$ in Ψ connects $\imath(e)$ with $\imath(f)$ and its label equals 1 in the free group. The lemma is proved. $\square$

Lemma 8.3 *Let Ψ be a non-reduced diagram over any presentation. Consider a process of cancelling a sequence of reducible pairs of cells in Ψ. If two cells π and π' in Ψ cancel in this process then these cells correspond to mutually inverse relations and there exists a path p in Ψ connecting corresponding vertices in $\partial(\pi)$ and $\partial(\pi')$ whose label is equal to 1 in the free group.*

Proof. Indeed it is clear that π and π' correspond to mutually inverse relations. After a number of cancellations in Ψ an edge e on $\partial(\pi)$ is identified with the corresponding edge f on $\partial(\pi')^{-1}$. It remains to apply Lemma 8.2. $\square$

Lemma 8.4 *Let Ψ be a non-reduced diagram over the presentation $\mathcal{P}_N(\mathcal{S})$ without κ-annuli and without hubs. Suppose that each κ-band in Ψ is reduced. Consider a process of cancelling a sequence of reducible pairs of cells in Ψ. Then no κ-annuli can appear as a result of this process and no two κ-cells from the same κ-band cancel.*

Proof. If we cancel two non-κ cells then κ-bands do not change. Let π and π' be κ-cells which form a reducible pair in Ψ. Let $\mathcal{B}$ and $\mathcal{B}'$ be the maximal κ-bands which contain π and π' respectively. Since π and π' are not hubs, they contain two κ-edges each. Let e and f be the κ-edges in π and e' and f' are the κ-edges in π'. Notice that the start and end edges of $\mathcal{B}$ and $\mathcal{B}'$ belong to the contour of Ψ. Finally assume without loss of generality that the median of $\mathcal{B}$ (resp. $\mathcal{B}'$) first crosses e and then f (resp. e' and f'). When we cancel π and π', the κ-edges in π are identified with the corresponding κ-edges in π'. By replacing $\mathcal{B}$ and $\mathcal{B}'$ with their inverses and renaming the edges e, f, e', f' if necessary, we can assume that e corresponds to e' and f corresponds to

f'. Let $\mathcal{B} = (\pi_1, \ldots, \pi_i, \pi, \pi_{i+1}, \ldots, \pi_t)$, $\mathcal{B}' = (\pi'_1, \ldots, \pi_j, \pi', \pi_{j+1}, \ldots, \pi_s)$. Then it is easy to check that after the cancellation we shall have two κ-bands $(\pi_1, \ldots, \pi_i, \pi_j, \pi_{j-1}, \ldots, \pi_1)$ and $(\pi_t, \ldots, \pi_{i+1}, \pi_{j+1}, \ldots, \pi_s)$ (and their inverses) instead of $\mathcal{B}$ and $\mathcal{B}'$ (one or both of these bands may be empty). Both these κ-bands start and end on the contour of Ψ. All other maximal κ-bands do not change. Therefore we do not get κ-annuli by reducing π and π'.

Now suppose that two cells π and π' of Ψ from the same κ-band $\mathcal{B}$ eventually cancel as a result of our process of cancellation. Then by Lemma 8.3 these cells correspond to mutually inverse relations and there exists a path t in Ψ connecting corresponding vertices v and v' of these cells whose label is 1 in the free group. It is easy to see that v and v' both belong to either $\mathbf{top}(\mathcal{B})$ or to $\mathbf{bot}(\mathcal{B})$. Therefore they are connected by a path t' which is contained in $\mathbf{top}(\mathcal{B})$ or $\mathbf{bot}(\mathcal{B})$. Since t and t' connect the same vertices in Ψ, their labels must be the same modulo the presentation obtained from $\mathcal{P}_N(\mathcal{S})$ by removing the hub. Therefore $\mathrm{Lab}(t') = 1$ modulo this presentation. But $\mathrm{Lab}(t')$ is a word over Θ. By Lemma 7.3, this word must be equal to 1 in the free group. Therefore $\mathbf{top}(\mathcal{B})$ contains two consecutive edges whose labels are mutually inverse. Then the corresponding cells form a reducible pair. This contradicts the assumption that every κ-band in Ψ is reduced. The lemma is proved. $\Box$

9 Sectors

By a *sector* we mean a reduced diagram Δ over $\mathcal{P}$ with boundary divided into four parts, $\partial(\Delta) = p_1 p_2 p_3^{-1} p_4^{-1}$, such that the following properties hold:

- p_1 is the top path of an odd κ-band $\mathcal{K}$;
- p_3 is the bottom path of an even κ-band $\mathcal{T}$;
- p_2 is the top path of a Θ-band $\mathcal{R}_t$, and the label $\mathrm{Lab}(p_2)$ is a reduced word;
- $\mathrm{Lab}(p_4) = \kappa_j^{\pm 1} W \kappa_s^{\pm 1}$ where W is an admissible word for which there exists a computation connecting W with W_0 (see Figure 5).

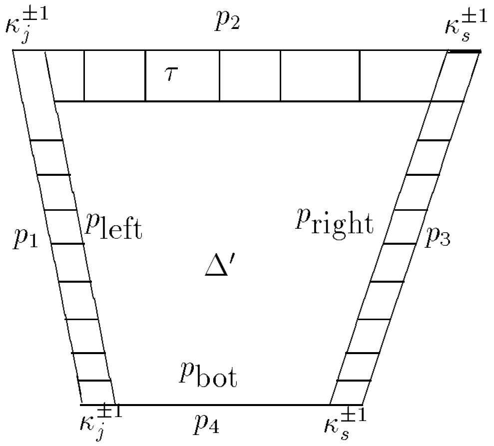

Fig. 5.

Recall that a κ_j-band is called *even* (resp. *odd*) if j is even (resp. odd).

If Δ is a sector then by removing the bands $\mathcal{K}$, $\mathcal{T}$ we get the *inside* diagram Δ' which has a boundary divided into 4 parts p_{left}, p_{top}, p_{right}^{-1}, p_{bot}^{-1} such that $p_{\text{left}} \subseteq \mathbf{bot}(\mathcal{K})$, $p_{\text{top}} \subseteq p_2$, $p_{\text{right}} \subseteq \mathbf{top}(\mathcal{T})$, $p_{\text{bot}} \subseteq p_4$.

In this section, we shall provide a complete description of sectors.

We start with examples of sectors which correspond to computations of our machine $\mathcal{S}$.

As in the definition of the hub, for every word u let

$$K(u) \equiv (u^{-1}\kappa_1 u\kappa_2 u^{-1}\kappa_3 u\kappa_4 \dots u^{-1}\kappa_{2N-1}u\kappa_{2N})(\kappa_{2N}u^{-1}\kappa_{2N-1}u \dots \kappa_2 u^{-1}\kappa_1 u)^{-1}.$$

Notice that if $u = W_0$ then $K(u)$ is the boundary label of the hub.

With every computation $W_1, \dots, W_g$ we associate a van Kampen diagram over the presentation $\mathcal{P}_N(\mathcal{S})$ in the following way.

If W_1, W_2 is a one-step computation which uses a transition $\tau \in \Theta$ then there exists an annular diagram with boundary labels $K(W_1)$ and $K(W_2)$ of the form shown in Figure 6(in the case when $N = 1$):

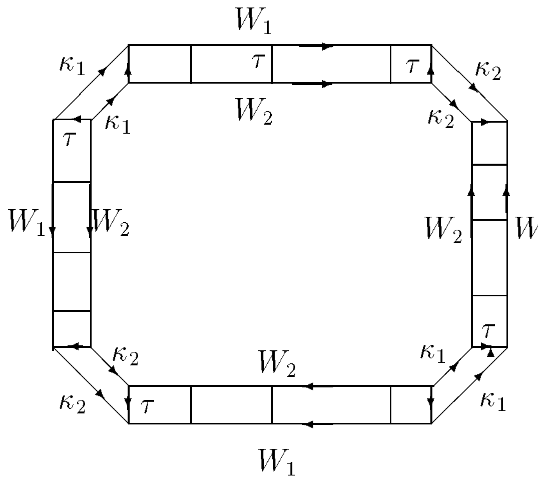

Fig. 6.

This Θ-annulus is composed of cells corresponding to relations involving τ. If we remove the $4N$ corners (κ-cells) then the annulus breaks into $4N$ equal diagrams. Each of them is obtained in the following way. Consider a path p labelled by W_1. Let $\tau = [U_1 \to V_1, U_2 \to V_2, ...]$. Since τ is applicable to W_1, this word contains all U_i's. Attach transition cells corresponding to the relations $U_i^\tau = V_i$ to the subpaths of p labelled by U_i. Attach auxiliary τ-cells to the remaining edges of p. Finally fold all pairs of edges with the same label and the same initial vertex. It is easy to see that the resulting diagram is a Θ-band whose common edges are labelled by τ, the top path is labelled by W_1 and the bottom path is labelled by W_2.

Consider now an arbitrary computation $C = (W_1, W_2, \ldots, W_g)$ with a history word h. Suppose that $\mathcal{CS}(W_g, W_0)$ is not empty. With every $i = 1, \ldots, g-1$ we associate the annular diagram with boundaries $K(W_i)$ and $K(W_{i+1})$ as in Fig. 6. Since for every $i = 1, \ldots, g-1$ the inner contour of the i-th annular diagram is labelled by the same word as the outer contour of the $(i+1)$-st diagram, we can "concatenate" all these diagrams by gluing the next diagram in the hole of the previous diagram. As a result we obtain an annular diagram with outer contour labelled by $K(W_1)$ and the inner contour labelled by $K(W_g)$. This diagram is called a *computational annulus* corresponding to the computation $W_1, ..., W_g$. This annular diagram is a union of $4N$ sectors and inverses of sectors which will be called *computational sectors corresponding to the computation C*. Notice that the history word h of the computation C is equal to the label of the top (and bottom) path of any maximal κ-band starting on the outer contour of the annulus.

If C is an accepting computation then the inner boundary of the annular diagram corresponding to C is the boundary of the hub. Thus we can glue in the hub and obtain an (ordinary) van Kampen diagram with boundary label $K(W_1)$. We call it the *computational disk corresponding to the computation C*.

It is easy to observe that the area of the computational sector corresponding to the computation C is $O(\text{area}(C))$ where $\text{area}(C)$ is the area of the computation C. The area of a computational disc corresponding to C is thus also equal to $O(\text{area}(C))$. We shall give an "abstract" definition of computational discs in the next section.

In this section we shall prove that every sector can be transformed into a computational sector without changing the boundary label or increasing the area.

Consider an arbitrary sector Δ with the inside diagram Δ', with boundary

$$\partial(\Delta') = p_{\text{left}} p_{\text{top}} p_{\text{right}}^{-1} p_{\text{bot}}^{-1},$$

with an odd κ-band $\mathcal{K}$ and an even κ-band $\mathcal{T}$. We assume that Δ has the smallest area among all sectors with the same boundary label.

Notice that p_{left} and p_{right} consist of Θ-edges. The maximal Θ-bands starting on p_{left} must end on p_{right} since Δ has no (k,Θ)-annuli. For the same reason, maximal Θ-bands starting on p_{right} must end on p_{left}. Therefore the maximal Θ-bands establish a one-to-one correspondence between edges on p_{left} and edges on p_{right}. Since all Θ-edges in any Θ-band have the same label, the labels of p_{left} and p_{right} are the same. Let us denote the maximal Θ-bands in Δ starting on p_{left} by $\mathcal{R}_1$,...,$\mathcal{R}_\ell$ counting from bottom, p_{bot}, up to p_{top}.

For every $1 < i \leq \ell$ the path $\mathbf{bot}(\mathcal{R}_i)\mathbf{top}(\mathcal{R}_{i-1})^{-1}$ bounds a subdiagram of Δ. The label of this path does not contain Θ-edges. Therefore by Lemma 7.7 this subdiagram contains no cells. Hence $\mathbf{bot}(\mathcal{R}_i) = \mathbf{top}(\mathcal{R}_{i-1})$.

By the definition of a sector, the label of the bottom path of $\mathcal{R}_1$, that is the label of p_{bot}, is an admissible word W such that $\mathcal{CS}(W, W_0)$ is not empty. Therefore the label of $\mathbf{bot}(\mathcal{R}_1)$ is a reduced word. The label of $\mathbf{top}(\mathcal{R}_\ell)$ is a reduced word by the definition of a sector. We can also assume that the labels of $\mathbf{bot}(\mathcal{R}_i)$, $i = 1, ..., \ell$, are reduced words.

Indeed, if the label of the path $\mathbf{bot}(\mathcal{R}_i)$ is not reduced then it contains a subpath e_1e_2 where e_1, e_2 are edges labelled by a and a^{-1} respectively, where a is a letter. These two edges must belong to two different cells π_1 and π_2 of $\mathcal{R}_i$ as in the diagram on the left in the Figure 7.

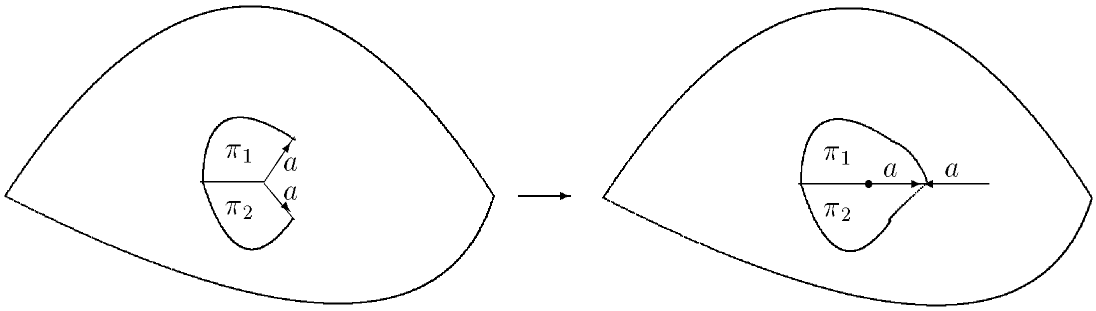

Figure 7.

Then we fold e_1 and e_2, producing a new edge e labelled by a and we introduce a new edge f with label a which has a common end vertex with e so that cells π_1 and π_2 have a common edge e. The other cells which were attached to e_1 and e_2 will be attached to the edge f. This operation does not change the area or the boundary of the diagram Δ. The new diagram $\bar{\Delta}$ must be reduced, otherwise after reducing it, we shall get a sector with a smaller area and the same boundary label as Δ (but we assumed that Δ has the smallest possible area). Thus the diagram $\bar{\Delta}$ is again a sector with the same number of maximal Θ-bands starting on p_{left}. The bottom path of the Θ-band $\mathcal{R}_i$ in $\bar{\Delta}$ is shorter (by two edges) than the bottom path of $\mathcal{R}_i$ in Δ. The bottom paths of the other Θ-bands are not affected by this operation. Thus after a finite number of such operation we get a sector with the same boundary label as Δ, in which all bands $\mathcal{R}_i$ have top and bottom paths with reduced labels.

The contour of Δ' does not have κ-letters. Since Δ does not have κ-annuli (this diagram is reduced), Δ' does not have κ-edges. In particular, it has no hubs.

For every component Q_i of the vector of sets of state letters Q there exists exactly one Q_i-edge on p_{bot} (here we use the form of the word W_0). Therefore Δ contains exactly one maximal Q_i-band starting on p_{bot} for every i. These Q_i-bands do not intersect and cannot end on p_{bot}, p_{left} or p_{right} (p_{left} and p_{right} contain no Q-letters). They cannot end on the contour of a hub because Δ has no hubs. Therefore these Q_i-bands end on p_{top}. Since Δ contains no (Q_j, Θ)-annuli (Lemma 7.4), each Q_i-band starting on p_{bot} intersects each Θ-band $\mathcal{R}_j$ $(j = 1, ..., \ell)$ exactly once. Also this implies that there are no Q_j-bands which start and end on p_{top}. Therefore for every component Q_i of the vector Q, Δ has exactly two mutually inverse maximal Q_i-bands. These Q_i-bands connect p_{top} and p_{bot} and intersect each of the Θ-bands $\mathcal{R}_j$ exactly once $(j = 1, ..., \ell)$.

Therefore each of the Θ-bands $\mathcal{R}_j$ contains exactly one Q_i-cell for every component Q_i.

Consider one of the Θ-bands $\mathcal{R} = \mathcal{R}_j$, $j = 1, ..., \ell$. Let W' be the label of the path $\mathbf{bot}(\mathcal{R}_j)$ and let $\tau \in \Theta$ be the label of the start edge of this Θ-band. Suppose that W' is an admissible word. Since $\mathcal{R}$ starts on p_{left}, the start edge of $\mathcal{R}$ belongs to p_{left}, so it is oriented toward p_{top}.

For convenience suppose that $\tau \in \Theta_+$ (the case when $\tau^{-1} \in \Theta_+$ is similar). Let

$$\tau = [U_1 \to V_1, U_2 \to V_2, ..., U_m \to V_m]$$

be the form of the S-rule τ. By the definition of S-rules, U_i starts and ends with state letters, and the state letters from different U_i's belong to different components of Q, so these state letters are different.

Let q and q' be the first and the last state letters of U_i, $q \in Q_a$, $q' \in Q_b$ for some numbers a and b, $a < b$. There exists only one (up to cyclic shifts and inverses) relation in $\mathcal{P}_N(\mathcal{S})$ containing a letter from Q_a and τ. This is the relation $U_i^\tau V_i^{-1} = 1$. The same relation is the only relation containing a letter from Q_b. We have proved that the Q_a-band and the Q_b-band starting on p_{bot} intersect with $\mathcal{R}$. Therefore their medians intersect with $\mathbf{bot}(\mathcal{R})$. Thus $\mathbf{bot}(\mathcal{R})$ contains a Q_a-edge e and a Q_b-edge e'. These edges are oriented from p_{left} to p_{right}, that is both of these edges belong to the path $p_{\text{bot}}(\mathcal{R})$. The labels of e and e' are q and q', respectively because these are the only Q_a- and Q_b-letters appearing in relations of $\mathcal{P}_N(\mathcal{S})$ together with τ. Let π and π' be cells in $\mathcal{R}$ containing e and e'. The boundary of π must have the form $e^{-1}tw'(t')^{-1}f^{-1}w^{-1}$ where $\text{Lab}(t) = \text{Lab}(t') = \tau$, $\text{Lab}(w') = V_i$, $\text{Lab}(f) = q'$, $\text{Lab}(ewf) = U_i$. If f is not equal to e' then f must belong to the contour of the a cell π'' of $\mathcal{R}$ which goes after π. Then f, e', the second Q_b-edge of π and the second Q_b-edge of π'' belong to the same Q_b-band (as we know there are only two mutually inverse Q_b-bands in Δ). Therefore this Q_b-band intersects $\mathcal{R}$ twice which is impossible by Lemma 7.4. Therefore $f = e'$. Thus W' contains subword U_i for every $i = 1, ..., m$, $\mathcal{R}$ has cells with boundary words $\tau V_i \tau^{-1} U_i^{-1}$ for every $i = 1, ..., m$, and the subpaths of these boundaries labelled by U_i^{-1} are subpaths of p_{bot}^{-1}. Notice that the subpaths labelled by V_i are the top paths of these cells considered as Θ-bands.

If W' has a Q_j-edge e which does not belong to any of the U_i then the cells involving e must correspond to the commutativity transition relation $q^\tau = q$ where $q = \text{Lab}(e)$.

Removing from $\mathcal{R}$ all these transition cells which were discussed in the last two paragraphs, we obtain several Θ-bands without Q-edges on their contours. By Lemma 7.2 these Θ-bands must consist of auxiliary cells. It is clear that such a Θ-band is uniquely determined by the label of its bottom path (which is equal to the label of its top path because the top and the bottom paths of an auxiliary cell considered as a Θ-band, coincide).

Summarizing the information that we have obtained, we can conclude that W' has the form

$$g_1 P_1 g_2 P_2 ... g_c P_c g_{c+1}$$

where g_i are words without state letters, and either $P_i = U_j$ or $P_i \in Q_j$ such that none of the U's contain a Q_j-letter. The label of the top path of $\mathcal{R}$ is obtained by reducing the word

$$g_1 R_1 g_2 R_2 ... g_c R_c g_{c+1}$$

where $R_i = V_i$ if $P_i = U_i$ and $R_i = P_i$ if P_i is none of the U's. Thus the label of $\mathbf{top}(\mathcal{R})$ is obtained from W' by applying the S-rule τ.

Notice that if the start and end edges of $\mathcal{R}$ are labelled by τ^{-1} where $\tau \in \Theta_+$ then the same argument shows that $\mathrm{Lab}(\mathbf{top}(\mathcal{R}))$ is obtained by applying the rule τ^{-1} to W'.

We have proved that the sequence C of words

$$\mathrm{Lab}(p_{\mathrm{bot}}), \mathrm{Lab}(\mathbf{top}(\mathcal{R}_1)), \mathrm{Lab}(\mathbf{top}(\mathcal{R}_2)), ..., \mathrm{Lab}(\mathbf{top}(\mathcal{R}_\ell))$$

is a computation of the S-machine $\mathcal{S}$.

Notice also that our argument shows that there is a unique Θ-band with top path labelled by $\mathrm{Lab}(\mathbf{top}(\mathcal{R}))$, the bottom path labelled by W' and the Θ-edges labelled by τ. In order to create this band one needs to draw a path p labelled by W', glue in cells with boundary labels $\tau V_1 \tau^{-1} U_1^{-1}$, $\tau V_2 \tau^{-1} U_2^{-1}$, ..., $\tau V_m \tau^{-1} U_m^{-1}$ to the parts of the path p labelled by U_1, ..., U_m respectively. Then attach auxiliary τ-cells to the edges of p which are not in the parts of p which we used before. If after that we get two edges with the same labels and the same initial vertex, we fold them. After all these foldings we get a Θ-band with the top path labelled by $\mathrm{Lab}(\mathbf{top}(\mathcal{R})$ and the bottom path labelled by W'.

Therefore the sector Δ coincides with the computational sector corresponding to the computation C. Thus we have proved the following result.

Proposition 9.1 *Every sector can be transformed into a computational sector without increasing the area or the boundary label. The area of a computational sector is "big O" of the area of the corresponding computation*

10 Computational Discs

Let Δ be a reduced diagram over $\mathcal{P}_N(\mathcal{S})$ with reduced boundary label and exactly one hub. Suppose further that Δ does not have Θ-edges on the contour. In this case we call Δ a *disc*. The goal of this section is to provide a description of discs.

Lemma 10.1 *If Δ is a disc then every maximal κ-band in Δ contains the hub, and every maximal Q-band starts or ends on the contour of the hub.*

Proof. Suppose that there exists a maximal κ-band $\mathcal{K}$ which does not contain the hub. By Lemma 7.1, $\mathcal{K}$ starts and ends on the contour of Δ. Cutting Δ along the path $\mathbf{top}(\mathcal{K})$ we obtain two diagrams Δ_1 and Δ_2. Since Δ has only one hub, one of these diagrams does not contain a hub. Without loss of generality assume that Δ_1 is hub-free. The path $\mathbf{top}(\mathcal{K})$ consists of Θ-edges since $\mathcal{K}$ does not contain hubs. These Θ-edges cannot cancel when we form $\mathbf{top}(\mathcal{K})$, otherwise $\mathcal{K}$ would not be reduced. Consider the maximal Θ-bands in Δ_1 starting on $\mathbf{top}(\mathcal{K})$. Since Δ_1 does not contain hubs, these Θ-bands cannot end on $\mathbf{top}(\mathcal{K})$ (Lemma 7.5). Therefore they end on $\partial(\Delta)$. This contradicts the assumption that $\partial(\Delta)$ does not contain Θ-edges.

If Δ has a maximal Q-band $\mathcal{Q}$ which does not start or end on the contour of the hub then $\mathcal{Q}$ starts and ends on the contour of Δ (Lemma 7.2). Cutting Δ along $\mathbf{top}(\mathcal{Q})$ we obtain two diagrams Δ_1 and Δ_2, one of which, say Δ_1, does not contain hubs. The

Θ-edges do not cancel when we form the path $\mathbf{top}(\mathcal{Q})$ because otherwise two cells in $\mathcal{Q}$ form a reducible pair. Therefore the path $\mathbf{top}(\mathcal{Q})$ contains Θ-edges. By Lemma 7.4 the maximal Θ-band starting on one of these edges must end on $\partial(\Delta)$. This again contradicts the assumption that $\partial(\Delta)$ does not have Θ-edges. The lemma is proved. $\square$

Suppose that Δ is a disc. Since every cell in Δ (except the hub) contains a Θ-edge, Δ is covered by Θ-bands (except for the hub). Since there are no Θ-edges on the contour of Δ, each maximal Θ-band in Δ is an annulus. Every Θ-annulus must contain a hub in its inside diagram (Lemma 7.6). Therefore every Θ-annulus in Δ goes around the hub, so the Θ-annuli in Δ form concentric annuli surrounding the hub. We shall consider only the Θ-annuli that go clockwise around the hub (other Θ-annuli are the inverses of these Θ-annuli). Let $\mathcal{R}_o$ be the outermost Θ-annulus and $\mathcal{R}_i$ be the innermost Θ-annulus. Every edge of the contour of Δ belongs to the contour of a cell. Indeed, otherwise by removing this edge one gets two disjoint diagrams one of which does not contain hubs. Since this hub-free diagram does not have Θ-edges on its contour, it must be empty (Lemma 7.7). Since the Θ-bands in Δ form concentric rings, all contour edges must belong to contours of cells from $\mathbf{top}(\mathcal{R}_o)$. It is clear that $\mathbf{bot}(\mathcal{R}_i)\cap\mathbf{bot}(\mathcal{R}_i)^{-1}$ is the contour of the hub (there cannot be any cells between $\mathbf{bot}(\mathcal{R}_i)$ and the contour of the hub).

Let us remove the hub π from Δ and consider the maximal κ-bands in the resulting annular diagram Δ_1 starting on $\partial(\pi)$. Let us enumerate these bands clockwise by $\mathcal{B}_1,...,\mathcal{B}_{4N}$. Since Δ contains no maximal κ-bands which do not contain the hub (Lemma 10.1), and since there are no κ-annuli in Δ (Lemma 7.1), Δ_1 is covered by $4N$ subdiagrams Σ_1,..., Σ_{4N} such that Σ_i is bounded by $\mathbf{top}(\mathcal{B}_i)$, a part of $\partial(\Delta)$, $\mathbf{bot}(\mathcal{B}_{i+1})$ (addition modulo $4N$) and a part of $\partial(\pi)$. By definition each of Σ_i is either a sector or the inverse of a sector, depending on whether the κ-band $\mathcal{B}_i$ is even or odd; $\mathcal{R}_o\cap\Sigma_i$ is the top Θ-band in this sector. Using Proposition 9.1, we can transform each sector $\Sigma_i^{\pm1}$ into a computational sector of the same or smaller area and with the same boundary. Thus we can assume that each Σ_i is a computational sector.

The label of p_{bot} in each of these sectors is W_0. Let w be the history word of the sector Σ_1. This word is equal to $\mathrm{Lab}(\mathbf{top}(\mathcal{B}_1))$. By the definition of computational sectors,

$$\mathrm{Lab}(\mathbf{top}(\mathcal{B}_i)) = \mathrm{Lab}(\mathbf{bot}(\mathcal{B}_{i+1}) = \mathrm{Lab}(\mathbf{top}(\mathcal{B}_{i+1}))$$

for every $i=1,...,4N$. Therefore the history words of all these sectors are the same. By the definition of a computational sector, computational sectors with the same label of the bottom paths and the same history words are homotopic. This implies that if we let $\mathrm{Lab}(\mathbf{top}(\mathcal{R}_o\cap\Sigma_1))=u$, then

$$\mathrm{Lab}(\partial(\Delta)) = K(u).$$

It is clear that the area of the disc is "big O" of the area of the corresponding computation. Let us compute the diameter of the disc. Take any vertex v inside the disc. It belongs to one of the Θ-annuli of the disc. Let is be the Θ-band number i counted from the hub to the boundary of the disc. It is clear that this vertex is within a constant distance (less than the maximal length of a relator of $G_N(\mathcal{S})$) from the Θ-band $i-1$. Therefore v is within distance Ci from the boundary of the disc. Since the number of Θ-annuli in the disc is equal to the length of the corresponding computation, the diameter of the disc is bounded above by a constant multiple of this length. On the other hand pick a vertex on the hub and consider any path p connecting this vertex with a vertex on the boundary of the disc. By Jordan's lemma, this path must cross medians of all Θ-annuli in the disc. Therefore the length of p cannot be smaller than the length of the computation corresponding to the disc.

This proves the following statement.

Proposition 10.1 *Every disc can be transformed without changing the boundary and without increasing the area into a computational disc. The area of this disc is "big O" of the area of the corresponding computation. The diameter of the disc is "big O" of the time of the corresponding computation.*

11 The Upper Bound

In this section, we shall prove that for every word w which is equal to 1 in $G_N(\mathcal{S})$ there exists a van Kampen diagram Δ over $\mathcal{P}_N(\mathcal{S})$ with $w = \mathrm{Lab}(\partial(\Delta))$ such that Δ can be decomposed in some standard way into diagrams with at most one hub (like a snowman can be decomposed into small snow balls). This decomposition will then help us find an upper bound for the area and diameter of Δ, as functions of $|w|$. We assume that the function $T^4(|w|)$ (the area function of the machine $\mathcal{S} = \mathcal{S}(M)$ is superadditive.

Consider the following graph $G(\Delta)$ associated with the diagram Δ. The vertices of Δ are the hubs and the edges are maximal hub-free κ-bands connecting the hubs. Since κ-bands cannot intersect except in the hubs, $G(\Delta)$ is a plane graph (map). Every internal vertex of this graph has degree $4N$. We use the theory of plane graphs (maps) similar to small cancellation theory (see [20]). In order to do that we need the following fact.

Lemma 11.1 *If Δ has the smallest possible number of hubs among all reduced diagrams with the same boundary label, then $G(\Delta)$ does not contain polygons with only one vertex and does not have bigons (polygons with only two vertices).*

Proof. The first part of the statement follows immediately from the fact that Δ does not have κ-annuli.

Suppose that Δ has a bigon with two vertices π_1 and π_2 and two edges $\mathcal{B}_1$ and $\mathcal{B}_2$. We can assume that there are no κ-bands inside the bigon between $\mathcal{B}_1$ and $\mathcal{B}_2$, so (considered as edges of $G(\Delta)$), $\mathcal{B}_1$ and $\mathcal{B}_2$ are consecutive edges of π_1.

This implies that one of the bands $\mathcal{B}_1$ or $\mathcal{B}_2$ is a κ_j-band and another one is a κ_{j+1}-band. Let p_2 (resp. p_4) be the shortest subpath of $\partial(\pi_1)$ (resp. $\partial(\pi_2)$) containing the start (resp. the end) edges of $\mathcal{B}_1$ and $\mathcal{B}_2$. Then $\mathrm{Lab}(p_2) = \mathrm{Lab}(p_4)^{-1} = (\kappa_j W_0 \kappa_{j+1})^{\pm 1}$. Without loss of generality we can assume that $\mathrm{Lab}(p_2) = \mathrm{Lab}(p_4)^{-1} = \kappa_j W_0 \kappa_{j+1}$. By renaming $\mathcal{B}_1$ and $\mathcal{B}_2$ and by replacing Δ by Δ^{-1} if necessary we can also make the path $\mathbf{top}(\mathcal{B}_1) p_2 \mathbf{bot}(\mathcal{B}_2)^{-1} p_4^{-1}$ bound a subdiagram Σ containing the bands $\mathcal{B}_1$ and $\mathcal{B}_2$. Every Θ-band which crosses $\mathcal{B}_1$ must cross $\mathcal{B}_2$ (Lemma 7.5). Therefore p_2 is the top path of a Θ-band in Σ (the one which is the closest to π_1). Thus Σ is a sector. By Proposition 9.1, we can assume that Σ is a computational sector. Let $\partial(\pi_2) = p_4 p_5$.

Let w be the history word for Σ. Let D be the computational disc with a hub π corresponding to the same history word. Then D contains a copy Σ_1 of Σ. The boundary of the diagram $D_1 = D \backslash (\Sigma_1 \cup \pi)$ has the form $s_1 s_2$ where

$$\mathrm{Lab}(s_2) = \mathrm{Lab}(\mathbf{top}(\mathcal{B}_1) p_5 \mathbf{bot}(\mathcal{B}_2)^{-1}).$$

Thus we can cut Δ along the path $\mathbf{top}(\mathcal{B}_1) p_5 \mathbf{bot}(\mathcal{B}_2)^{-1}$ and insert the diagram

$$D_1 \circ_{s_1 = s_1} D_1^{-1}$$

in the hole (see Fig. 8).

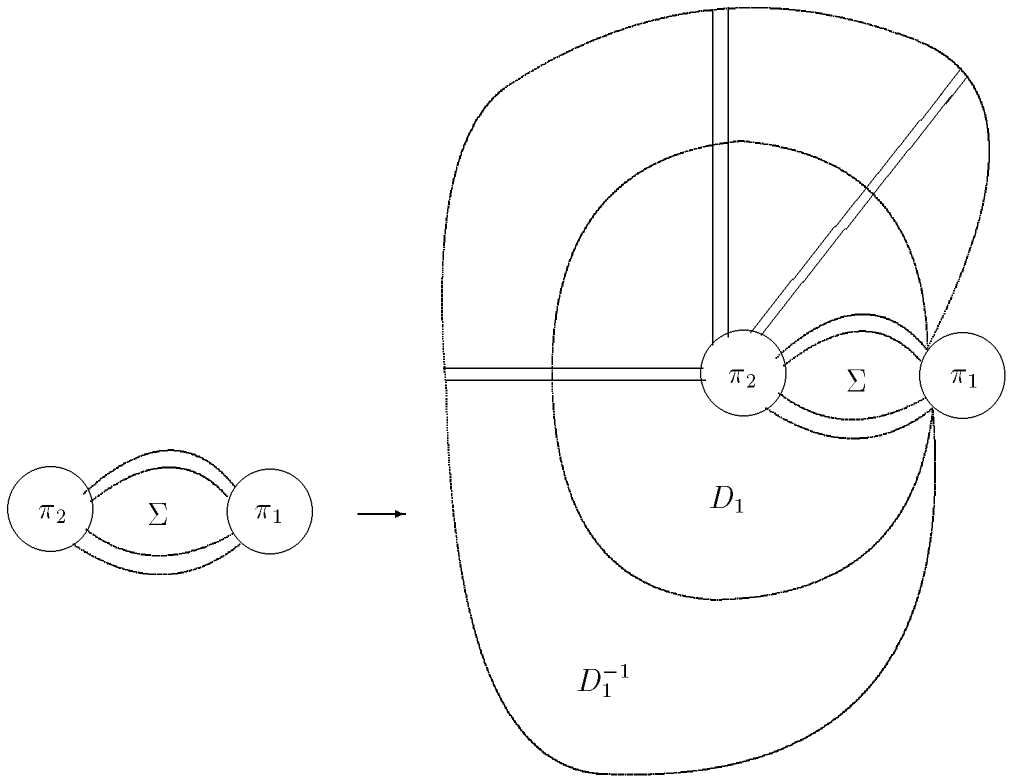

Fig. 8.

The resulting diagram will have the same boundary label as Δ. The maximal disc with the hub π_2 in this diagram corresponds to the history word w. Thus the boundary label of this disc will be $K(W_0)$, the same as the boundary label of the hub π_2. Therefore this disc without the hub is an annulus whose inner and outer boundaries have the same labels. Thus we can remove this annulus and obtain a diagram Δ' with the same boundary label and the same number of hubs as Δ (see the proof of Lemma 7.1). But in Δ', the hubs π_1 and π_2 have a common path labelled by $\kappa_j W_0 \kappa_s$. By Lemma 5.1 (d), these two cells cancel. By cancelling these hubs we obtain a diagram with the same boundary label and a smaller number of hubs. This contradicts the assumption that Δ contains the minimal possible number of hubs. The lemma is proved. $\square$

Lemma 11.2 $G(\Delta)$ *does not have triangular faces.*

Proof. Indeed, suppose that there are three hubs π_1, π_2, π_3 connected by three bands $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ such that $\mathcal{B}_2$ and $\mathcal{B}_3$ start on the contour of π_1 and end on the boundaries of π_3 and π_2 respectively, $\mathcal{B}_1$ connects boundaries of π_2 and π_3, none of the bands $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ contains hubs, and there are no hubs in the triangle bounded by these bands. Since $\mathcal{B}_2$ and $\mathcal{B}_3$ are two consecutive edges of π_1 in $G(\Delta)$, the κ-edges in one of these bands have even index and the κ-edges in the other band have odd index. Let, say, $\mathcal{B}_2$ be a κ_i-band with even i and $\mathcal{B}_3$ be a κ_j-band with odd j. The band $\mathcal{B}_1$ connects π_2 and π_3. Since $\mathcal{B}_1$ and $\mathcal{B}_3$ are consecutive edges of π_2, the index of κ-edges in $\mathcal{B}_1$ must be a even. But a similar argument applied to the hub π_3 gives that this index must be odd, a contradiction. $\square$

Lemmas 11.1 and 11.2 shows that all faces in the planar graph $G(\Delta)$ have degrees at least 4.

The following two lemmas are probably well known but we were not able to find them in the literature.

Lemma 11.3 *Let G be a non-empty plane graph in which every face has degree at least 4. Then G either consists of one vertex or has two vertices of degree at most 3.*

Proof. Suppose that G contains more than one vertex. We shall use the notation from [20] (Chapter 5). Let V, E, F be the numbers of vertices, edges and faces in G. Let Q be the number of connected components in G. Σ refers to the summation over all vertices v or faces D in G, $E^{\cdot}$ denotes the number of all boundary edges of G (the length of the boundary of G). By $d(v)$ and $d(D)$ we denote the degree of a vertex v or a face D. Then by Theorem 3.1 in Chapter 5 of [20] the following formula holds (we take $p = q = 4, h = 0$ in the formula in this theorem):

$$4Q = \Sigma[4 - d(v)] + \Sigma[4 - d(D)] - E^{\cdot}. \tag{29}$$

But by assumption, $d(D) \geq 4$ for every face D. Therefore $4Q < \Sigma[4 - d(v)]$. The left hand side is at least 4. Hence either there exists a vertex with $d(v) = 0$, or there are at least two vertices of degree at most 3. In the first case the vertex of degree 0 forms a connected component of the graph. Since G has more than one vertex, $Q \geq 2$. Then the left hand side of (29) is at least 8. This implies that the right hand side contains at least two positive terms, that is G contains two vertices of degree at most 3. $\square$

Now we are in a position to define the snowman decomposition of Δ. For the rest of this section we fix a large enough number N. Later we'll show that we can take $N \geq 6$ where k is the number of tapes of the original Turing machine. Recall that the number of sectors in every computational disc is $4N$.

With every diagram Δ we associate two numbers:

- $n(\Delta) = |\partial(\Delta)|$ — the length of the boundary of Δ;
- $h(\Delta)$ — the number of hubs in Δ.

Let $\mathcal{K}$ be a κ-band in Δ. Let Δ', Ψ be the subdiagrams obtained by dividing Δ along $\mathbf{top}(\mathcal{K})$ such that Ψ contains $\mathcal{K}$. If Ψ does not contain hubs then we call $\mathcal{K}$ a *dividing κ-band*. In particular a dividing κ-band is hub-free.

Suppose that Δ contains a dividing κ-band $\mathcal{K}$. Clearly both Δ' and Ψ are reduced, $h(\Delta) = h(\Delta')$. Suppose that $\partial(\Psi) = \mathbf{top}(\mathcal{K})p_1$, $\partial(\Delta') = p_2\mathbf{top}(\mathcal{K})^{-1}$. Since $\mathcal{K}$ does not contain hubs, the word $\mathrm{Lab}(\mathbf{top}(\mathcal{K}))$ consists of Θ-letters. The maximal Θ-bands in Ψ which start on $\mathbf{top}(\mathcal{K})$ cannot end on $\mathbf{top}(\mathcal{K})$ because of Lemma 7.5. Therefore all these Θ-bands end on p_1 (see Fig. 9). Since p_1 contains at least 2 κ-edges, we have $|p_1| \geq |\mathbf{top}(\mathcal{K})| + 2$. This implies that $|\partial(\Psi)| \leq 2|\partial(\Delta)|$ and $|\partial(\Delta')| \leq |\partial(\Delta)| - 2$.

Thus we have proved the following properties of the pair (Ψ, Δ').

(D1) Ψ and Δ' are reduced.

(D2) By gluing Ψ and Δ' we obtain a diagram with the same boundary label as Δ.

(D3) Ψ does not contain hubs.

(D4) Δ' contains a smaller number of dividing κ-bands than Δ.

(D5) $n(\Psi) \leq 2n(\Delta)$, $n(\Delta') \leq n(\Delta) - 2$, $h(\Delta') = h(\Delta)$.

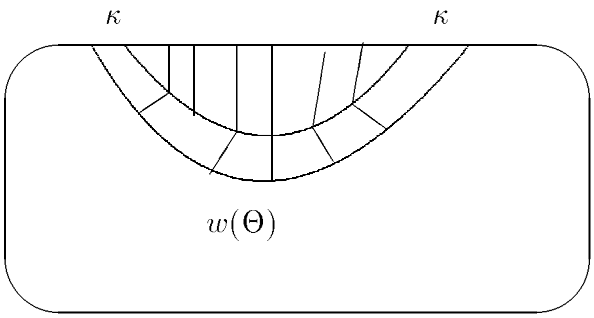

Figure 9.

In Figure 9 $w(\Theta)$ denotes any group word over the alphabet Θ.

Since Ψ has no hubs, its area can be computed with the help of Lemma 8.1. Thus in order to estimate the area of Δ we can replace Δ by Δ'. We shall call the pair (Ψ, Δ') the *decomposition of type 1 of* Δ.

Suppose that Δ contains no dividing κ-bands. Then by Lemma 11.3 there exists a vertex π in $G(\Delta)$ of degree at most 3. Let $\mathcal{B}_1, \ldots, \mathcal{B}_{4N}$ be the maximal κ-bands starting on the boundary of π, counted clockwise. Since the degree of π is at most 3, at least $4N - 3$ of these bands do not contain hubs (so they are not edges in $G(\Delta)$).

Lemma 11.4 *There exist at least two hubs π and π' in $G(\Delta)$ of degree at most 3 in $G(\Delta)$ such that $4N - 3$ consecutive κ-bands starting on $\partial(\pi)$ (resp. $\partial(\pi')$) do not contain hubs.*

Proof. Induction on the number of hubs. By Lemma 11.3, $G(\Delta)$ has at least two vertices π and π' of degree at most 3. Suppose that π does not have $4N-3$ consecutive κ-bands without hubs. Then there exist three maximal κ-bands $\mathcal{B}_i$, $\mathcal{B}_j$, $\mathcal{B}_m$, $i < j-1$, $j < m-1$ starting on $\partial(\pi)$ such that $\mathcal{B}_i$, $\mathcal{B}_j$ and $\mathcal{B}_m$ do not have hubs, but at least one band among $\mathcal{B}_{i+1}, \ldots, \mathcal{B}_{j-1}$ and at least one band among $\mathcal{B}_{j+1}, \ldots, \mathcal{B}_{m-1}$ contains a hub.

Consider the subdiagram Ψ of Δ containing $\mathcal{B}_i, \ldots, \mathcal{B}_{j-1}$ and π, and bounded by $\mathbf{top}(\mathcal{B}_i)$, $\mathbf{bot}(\mathcal{B}_{j-1})$, a part of $\partial(\pi)$ and a part of $\partial(\Delta)$. The subdiagram Ψ contains fewer hubs than Δ because among the κ-bands $\mathcal{B}_{j+1}, \ldots, \mathcal{B}_{m-1}$ there is at least one hub, which is not contained in Ψ. By the induction hypothesis, Ψ contains two hubs π_1 and π_2 of degree at most 3 such that there are $4N-3$ consecutive maximal κ-bands starting on π_1 (resp. π_2) which do not contain hubs. One of these hubs, say, π_1 differs from π. It is easy to see that π_1 considered as a vertex in $G(\Delta)$ also has degree at most 3 and $4N-3$ consecutive κ-bands without hubs.

Now consider the subdiagram Ψ' of Δ containing $\mathcal{B}_{j+1}, \ldots, \mathcal{B}_{m-1}$ and π bounded by $\mathbf{top}(\mathcal{B}_j)$, $\mathbf{bot}(\mathcal{B}_m)$, a part of $\partial(\pi)$ and a part of $\partial(\Delta)$. The same argument as in the case of the subdiagram Ψ shows that Ψ' contains a hub π_1' of degree at most 3 which has 4N-3 consecutive κ-bands without hubs. Thus we found two hubs π_1 and π_1' which satisfy the conditions of the lemma. $\Box$

Lemma 11.5 *For every hub π in Δ with hub-free consecutive maximal κ-bands $\mathcal{B}_1$,..., $\mathcal{B}_{4N-3}$ starting on $\partial(\pi)$, let $\Psi_\Delta(\pi)$ be the subdiagram of Δ bounded by* $\mathbf{top}(\mathcal{B}_1)$, $\mathbf{bot}(\mathcal{B}_{4N-3})$, *$\partial(\Delta)$ and $\partial(\pi)$, which contains $\mathcal{B}_1$, $\mathcal{B}_{4N-3}$ and does not contain π (there is only one subdiagram in Δ satisfying these conditions). Then there exists a hub π such that $\Psi_\Delta(\pi)$ does not contain hubs (see Fig. 10).*

Fig. 10.

Proof. Induction on the number of hubs in Δ. By Lemma 11.4, there exists a hub π in Δ with hub-free consecutive maximal κ-bands $\mathcal{B}_1, \ldots, \mathcal{B}_{4N-3}$ starting on $\partial(\pi)$. Suppose that $\Delta' = \Psi_\Delta(\pi)$ contains hubs. Notice that Δ' contains fewer hubs than Δ since Δ' does not contain π. By the induction hypothesis Δ' contains a hub π' such that $\Psi_{\Delta'}(\pi')$ does not contain hubs. Notice that none of the κ-bands in Δ starting on the contour of π' can intersect any of the κ-bands $\mathcal{B}_1,\ldots \mathcal{B}_{4N-3}$ because these κ-bands do not contain hubs. Therefore $\Psi_{\Delta'}(\pi') = \Psi_\Delta(\pi')$, so π' is a hub for which $\Psi_\Delta(\pi')$ is hub-free. The lemma is proved. $\Box$

By Lemma 11.5 we can find a hub π and $4N-3$ consecutive κ-bands starting in $\partial(\pi)$ which do not have hubs and moreover $\Psi_\Delta(\pi)$ is hub-free. We can assume that these κ-bands have indices $1, ..., 2N, 1, ..., 2N-3$ (otherwise we can rename κ's). We denote these $4N-3$ κ-bands by $\mathcal{B}_1', \ldots, \mathcal{B}_{4N-3}'$ (as usual we enumerate the κ-bands clockwise).

Let Π be the maximal disc in Δ with the hub π. By Proposition 10.1, we can assume that Π is a computational disc. We can also assume that Π corresponds to a minimal (with respect to area) accepting computation. By removing the interior of Π from Δ we get an annular diagram Ψ. Denote the parts of $\mathcal{B}'_i$ $(i = 1, ..., 4N)$ in Ψ by $\mathcal{B}_i$. Cutting Ψ along $\mathbf{top}(\mathcal{B}_1)$ and $\mathbf{bot}(\mathcal{B}_{4N-3})$ we obtain two ordinary van Kampen diagrams Ψ_1 and Ψ_2 such that $\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_{4N-3} \subset \Psi_1$, $\mathcal{B}_{4N-2}, \mathcal{B}_{4N-1}, \mathcal{B}_{4N} \subset \Psi_2$. Notice that $\Psi_1 = \Psi_\Delta(\pi) - \Pi$.

We shall fix the following notation associated with the triple (Ψ_1, Ψ_2, Π) (see Figures 11 and 12):

- $\partial(\Psi_1) = \mathbf{top}(\mathcal{B}_1) p_1 \mathbf{bot}(\mathcal{B}_{4N-3})^{-1} s_1$;
- $\partial(\Psi_2) = \mathbf{top}(\mathcal{B}_1)^{-1} s_2^{-1} \mathbf{bot}(\mathcal{B}_{4N-3}) p_2$;
- $\partial(\Pi) = s_1^{-1} s_2$;
- $p_1 = e(\mathcal{B}_1) t_1 e(\mathcal{B}_2) t_2 \ldots t_{4N-4} e(\mathcal{B}_{4N-3})$ where $e(\mathcal{B}_i)$ is the end edge of $\mathcal{B}_i$, t_j is a subpath of p_1, $i = 1, \ldots, 4N-3, j = 1, \ldots, 4N-4$;
-
$$s_1^{-1} = i(\mathcal{B}_1) y_1 i(\mathcal{B}_2) y_2 \ldots y_{4N-4} i(\mathcal{B}_{4N-3}),$$
$$s_2 = y_{4N-3} i(\mathcal{B}_{4N-2}) y_{4N-2} i(\mathcal{B}_{4N-1}) y_{4N-1} i(\mathcal{B}_{4N}) y_{4N}$$
where $i(\mathcal{B}_j)$ is the start edge of $\mathcal{B}_j$, for $j = 1, ..., 4N-4$, y_j is a subpath of s_1^{-1}, for $j = 4N-3, ..., 4N$, y_i is a subpath of s_2;
- Σ_i $(i = 1, \ldots, 4N-4)$ is a subdiagram of Ψ_1 bounded by $\mathbf{top}(\mathcal{B}_i)$ on the left, $\mathbf{bot}(\mathcal{B}_{i+1})$ on the right, p_1 on the top and s_1^{-1} on the bottom;
- if Ψ_1 contains a Θ-band $\mathcal{R}$ which crosses $\mathcal{B}_i$ and $\mathcal{B}_{i+1}$, and such that the intersection with $\mathcal{B}_i$ precedes in $\mathcal{R}$ the intersection with $\mathcal{B}_{i+1}$ then $t(\mathcal{R}, i)$ is the portion of $\mathbf{top}(\mathcal{R})$ which is contained in Σ_i; $t(\mathcal{R}, i) = f(\mathcal{R}, i) y(\mathcal{R}, i) f(\mathcal{R}, i+1)$ where $f(\mathcal{R}, i)$ and $f(\mathcal{R}, i+1)$ are κ-edges;
- $b = |\mathbf{top}(\mathcal{B}_1)| + |\mathbf{top}(\mathcal{B}_2)| + \ldots + |\mathbf{top}(\mathcal{B}_{4N-3})|$;
- $c = |y_1|$.

Fig. 11.

Fig. 12.

When we need to specify the triple $\mathcal{T} = (\Psi_1, \Psi_2, \Pi)$ to which this notation is associated, we shall write $\mathcal{B}_1(\mathcal{T})$ instead of $\mathcal{B}_1$, etc.

The triple (Ψ_1, Ψ_2, Π) satisfies the following properties.

(P1) By gluing together Ψ_1, Ψ_2 and Π one can get a diagram with the same boundary label as Δ.

(P2) Ψ_1 is reduced and contains no hubs. Π is a computational disc corresponding to a minimal area accepting computation.

(P3) The κ-bands $\mathcal{B}_i$, $i = 1, \ldots, 4N-3$, in Ψ_1 start on s_1^{-1}, and end on p_1. Every κ-band in Ψ_1 starts or ends on s_1.

(P4) The word $\mathrm{Lab}(f_1 y_1 \ldots f_{4N} y_{4N})$ is a cyclic shift of the word $K(W)$ for $W = \mathrm{Lab}(y_1)$. For all $i = 1, \ldots, 4N$, $c = |y_i|$.

(P5) $h(\Psi_2) + 1 = h(\Delta)$.

We assume from now on that we have a triple of diagrams (Ψ_1, Ψ_2, Π) which satisfies the properties (P1)–(P5). We shall transform this triple into another triple which satisfies the same properties plus some additional conditions.

Suppose that there exists a Θ-band in Ψ_1 which crosses the bands $\mathcal{B}_1, \ldots, \mathcal{B}_{4N-3}$. Then let $\mathcal{R}$ be the Θ-band with this property whose intersection with $\mathcal{B}_1$ is as far away from s_1 as possible (the distance is counted along $\mathcal{B}_1$). We assume that $\mathcal{R}$ starts on $\mathbf{top}(\mathcal{B}_1)$. Consider the diagram E bounded by $\mathbf{top}(\mathcal{B}_1)$, $\mathbf{top}(\mathcal{R})$, $\mathbf{bot}(\mathcal{B}_{4N-3})$ and s_1. Then $E' = E \circ_{s_1 = s_1} \cup_{i=1}^{4N-4} \Sigma_i'$ is a union of sectors and inverses of sectors which correspond to the same history word w'. Let us remove this subdiagram from Ψ_1, and consider the resulting diagram as a new Ψ_1. Then extend the disc Π by an annulus with the history word w' and replace this disc by a computational disc with the same boundary label which corresponds to a computation of $\mathcal{S}$ with the smallest possible area. Let it be the new Π. After that, glue Ψ_2 to the union of the four diagrams (sectors and inverse sectors with the history word w') Σ_i^{-1}, $i = 4N-3, 4N-2, 4N-1, 4N$, then reduce the resulting diagram. Let this diagram be the new Ψ_2 (see Fig. 13). The new triple (Ψ_1, Ψ_2, Π) obviously satisfies properties (P1)-(P5). We shall call this operation *moving Θ-bands*. In addition, this triple satisfies the following property.

(P6) There are no Θ-bands in Ψ_1 which cross all the bands $\mathcal{B}_1, \ldots, \mathcal{B}_{4N-3}$.

Fig. 13.

Now suppose that we have a triple $\mathcal{T} = (\Psi_1, \Psi_2, \Pi)$ which satisfies conditions (P1)–(P6).

Take any number $i = 1, \ldots, 4N-4$. Suppose that the following property *does not* hold:

(P7) There is no Θ-band $\mathcal{R}$ in Σ_i such that $|y(\mathcal{R}, i)| < |y_i|$.

Then take a Θ-band $\mathcal{R}$ such that $|y(\mathcal{R}, i)|$ (which is less than $|y_i|$) is the smallest possible (over all i and all Θ-bands $\mathcal{R}$).

Consider the subsector T of Σ_i bounded by $\mathbf{top}(\mathcal{B}_i)$, $t(\mathcal{R}, i)$, $\mathbf{bot}(\mathcal{B}_{i+1})$, $h_i y_i h_{i+1}$ (with $\mathcal{R}$ as the top Θ-band). Let w' be the history word of this sector. We can assume by Proposition 9.1 that T is a computational sector. Consider the computational annulus $\bar{E}$ corresponding to the history word w' and initial configuration $\mathrm{Lab}(y(\mathcal{R}, i))$. Then the disc Π fits in the hole of this annulus. By gluing Π inside $\bar{E}$, and replacing this computational disc by a disc corresponding to a minimal area computation, we get a computational disc which we denote by Π_1 (see Fig. 14).

Fig. 14.

Let E_1 be the union of $4N-4$ consecutive subsectors of $\bar{E}$ such that the part p of the inner boundary of $\bar{E}$ contained in E_1 has the same label as s_1. Let $E_2 = \bar{E} \backslash E_1$. Let us form a diagram $\Psi_{1,1}$ by the following procedure. First (*step 1*) we glue Ψ_1 and E_1^{-1} along $p = s_1$. Notice that the k-bands of E_1^{-1} will be glued to the κ-bands of Ψ_1 along the edges f_i, so the κ-bands are getting longer. Some κ-cells in Ψ_1 will cancel with κ-cell in E_1^{-1}. In *step 2* we cancel these pairs of κ-cells. Notice that if $j = 1, \ldots, 4N-3$ and w' has a common suffix with $\mathrm{Lab}(\mathbf{top}(\mathcal{B}_j))$ of length d_j then d_j cells of $\mathcal{B}_j$ will cancel. In particular, since w' is a suffix of $\mathrm{Lab}(\mathbf{top}(\mathcal{B}_i))$, $|w'|$ cells in $\mathcal{B}_i$ cancel. Then in *step 3* we reduce other reducible pairs of cells. The resulting (reduced) diagram is $\Psi_{1,1}$. Finally let us glue Ψ_2 and E_2^{-1}, reduce this diagram and denote the resulting diagram by $\Psi_{2,1}$. We call the process of obtaining the triple $(\Psi_{1,1}, \Psi_{2,1}, \Pi_1)$ an *adjustment*.

Lemma 11.6 *The triple $\mathcal{T}_1 = (\Psi_{1,1}, \Psi_{2,1}, \Pi_1)$ satisfies the properties (P1)–(P5). For every $j = 1, \ldots, 4N-3$, $\mathrm{Lab}(\mathbf{top}(\mathcal{B}_j(\mathcal{T}_1))) = (w')^{-1}\mathrm{Lab}(\mathbf{top}(\mathcal{B}_j(\mathcal{T})))$ (equality in the free group).*

Proof. Properties (P1) and (P2) are obvious.

After step 1 of building $\Psi_{1,1}$ the κ-bands of the resulting diagram will be just unions of corresponding pairs of κ-bands in Ψ_1 and E_1^{-1}, so they will start on $s_1(\mathcal{T}_1) \subseteq \partial(E_1^{-1})$ and end on $p_1(\mathcal{T}_1) = p_1(\mathcal{T})$. This property will still hold after step 2 because in this step we are cancelling cells from the same κ-bands (notice that $s_1(\mathcal{T}_1)$ and $p_1(\mathcal{T}_1)$ do not change after steps 2 and 3 because reducing a diagram does not affect its boundary). By Lemma 8.4, no κ-annuli are produced in step 3. So step 3 is just the process of cancelling reducible pairs of cells. By Lemma 8.4, no κ-cells from the same k-band cancel during this process. Cells from different κ-bands $\mathcal{B}_{j_1}$ and $\mathcal{B}_{j_2}$ do not form a reducible pair because these cells do not have common edges. Let Ψ be the diagram obtained after a number of reduction from step 3. Then for every $j = 1, \ldots, 4N-4$ there exists an $\mathbf{E}_0$-band which starts on $s_1(\mathcal{T}_1)$ between $\mathcal{B}_j$ and $\mathcal{B}_{j+1}$ and ends on $p_1(\mathcal{T}_1)$ between the same κ-bands. Therefore no cell in $\mathcal{B}_j$ can have a common edge with a cell in $\mathcal{B}_{j+1}$. Since Ψ does not have any other κ-bands, no κ-cells participate in the reduction process of step 3. Therefore the κ-bands $\mathcal{B}_j$, $j = 1, \ldots, 4N-3$, do not change during step 3. This proves property (P3) and the fact that in the free group, $\mathrm{Lab}(\mathbf{top}(\mathcal{B}_j(\mathcal{T}_1))) = (w')^{-1}\mathrm{Lab}(\mathbf{top}(\mathcal{B}_j(\mathcal{T})))$.

Properties (P4) and (P5) immediately follow from the definition of the triple $\mathcal{T}_1$. $\Box$

Since $\mathcal{T}_1$ satisfies properties (P1)–(P5) we can use all the notation associated with triples satisfying these properties. Notice that by construction $\mathcal{T}_1$ satisfies the property $c(\mathcal{T}_1) < c(\mathcal{T})$. The triple $\mathcal{T}_1$ may not satisfy property (P6). We shall discuss this property in the next lemma. Consider the following property:

(P6') The words $\mathrm{Lab}(\mathbf{top}(\mathcal{B}_j))$, $j = 1, \ldots, 4N-3$, do not all have the same first letter.

The following lemma shows, in particular, that (P6') is stronger than (P6).

Lemma 11.7 *a) If $\mathcal{T}_1$ does not satisfy property (P6') then w' is a prefix of the label of* $\mathbf{top}(\mathcal{B}_j(\mathcal{T}))$ *for every $j = 1, \ldots, 4N-3$. In this case*

$$b(\mathcal{T}_1) < b(\mathcal{T}).$$

b) If $\mathcal{T}_1$ satisfies property (P6') then $\mathcal{T}_1$ satisfies property (P6) (that is (P6') implies (P6)).

c) If $\mathcal{T}$ satisfies (P6') then so does $\mathcal{T}_1$.

Proof. a) If w' is not a prefix of $\mathrm{Lab}(\mathbf{top}(\mathcal{B}_j(\mathcal{T})))$ for some j then by Lemma 11.6 the first letter of $\mathrm{Lab}(\mathbf{top}(\mathcal{B}_j(\mathcal{T}_1)))$ is the first letter of $(w')^{-1}$. On the other hand, by construction, w' is a prefix of $\mathrm{Lab}(\mathbf{top}(\mathcal{B}_i(\mathcal{T})))$ (for the particular i considered above). Therefore the first letter in $\mathrm{Lab}(\mathbf{top}(\mathcal{B}_i(\mathcal{T}_1)))$ is the letter number $|w'|+1$ in $\mathrm{Lab}(\mathbf{top}(\mathcal{B}_i(\mathcal{T})))$. If these two first letters in $\mathrm{Lab}(\mathbf{top}(\mathcal{B}_j(\mathcal{T}_1)))$, resp. $\mathrm{Lab}(\mathbf{top}(\mathcal{B}_i(\mathcal{T}_1)))$ coincide, then $\mathrm{Lab}(\mathbf{top}(\mathcal{B}_i(\mathcal{T})))$ contains a pair of consecutive mutually inverse letters; but this is impossible since $\mathcal{B}_i(\mathcal{T})$ is reduced. Therefore if the first letter of each word $\mathrm{Lab}(\mathbf{top}(\mathcal{B}_j(\mathcal{T}_1)))$ is the same, then w' is a prefix of $\mathrm{Lab}(\mathbf{top}(\mathcal{B}_j(\mathcal{T})))$ for $j = 1, \ldots, 4N-3$. It is easy to see that in this case $b(\mathcal{T}_1) = b(\mathcal{T}) - (4N-3) * |w'|$.

b) If there exists a Θ-band in $\Psi_1(\mathcal{T}_1) = \Psi_{1,1}$ which crosses all the κ-bands $\mathcal{B}_i(\mathcal{T}_1)$, $i = 1, \ldots, 4N-3$, then there exists a Θ-band containing the first cells of all these κ-bands (this is the maximal Θ-band containing the first cell in $\mathcal{B}_1(\mathcal{T}_1)$), so the indices of the Θ-letters in the relations corresponding to these cells are the same. Therefore the first letters in the words $\mathrm{Lab}(\mathbf{top}(\mathcal{B}_i(\mathcal{T}_1)))$ are the same, a contradiction.

c) If the the words $\mathrm{Lab}(\mathbf{top}(\mathcal{B}_j(\mathcal{T})))$, $j = 1, \ldots, 4N-3$, do not all have the same first letter, then w' cannot be a prefix of all these words, and we can apply part (a). The lemma is proved. $\Box$

If $\mathcal{T}_1$ does not satisfy property (P6') then by Lemma 11.7, $b(\mathcal{T}_1) < b(\mathcal{T})$. If $\mathcal{T}_1$ does not satisfy (P6) then we can apply the band moving construction and obtain a triple $\mathcal{T}_2$ which satisfies (P1)–(P6). Notice that the band moving construction strictly decreases the parameter b. If $\mathcal{T}_2$ does not satisfy property (P7) then we can repeat the adjustment and band moving constructions and obtain a sequence of triples $\mathcal{T}_2$, $\mathcal{T}_3, \ldots$ such that $b(\mathcal{T}_1) > b(\mathcal{T}_2) > \ldots$. For some $s > 0$ the triple $\mathcal{T}_s$ satisfies condition (P6'). By Lemma 11.7, $\mathcal{T}_s$ satisfies condition (P6) and the triples obtained from $\mathcal{T}_s$ by any number of adjustments satisfy (P6'); therefore we do not need further applications of the band moving construction to obtain $\mathcal{T}_{s+1}, \mathcal{T}_{s+2}, \ldots$.

As we noticed before, $c(\mathcal{T}_1) < c(\mathcal{T})$. Since we do not use the band moving construction to obtain $\mathcal{T}_{s+1}, \ldots$, we have $c(\mathcal{T}_{s+1}) > c(\mathcal{T}_{s+2}) > \ldots$ (notice that the band moving construction can increase the parameter c). This sequence cannot be infinite so there exists t such that the triple $\mathcal{T}_t$ satisfies all seven properties (P1)–(P7). Without loss of generality we can assume now that $\mathcal{T}$ itself satisfies these properties.

Now let $\mathcal{T} = (\Psi_1, \Psi_2, \Pi)$ be a decomposition of Δ satisfying properties (P1)–(P7). We shall prove that the perimeters of Ψ_1, Ψ_2, Π relate nicely to the perimeter of Δ.

From now on we shall assume that

$$N \geq 6. \tag{30}$$

Notice that because of Property (P6) every Θ-band in Ψ_1 starting on $\mathbf{top}(\mathcal{B}_1) \cup \mathbf{bot}(\mathcal{B}_{4N-3})$ ends on p_1.

If a Θ-band $\mathcal{R}$ of Ψ_1 starting on an edge $e \in t_i$, $i \leq 4N-4$, ends on the contour of $\mathcal{B}_{4N-3}$ (resp. $\mathcal{B}_1$) then we paint the edge e in *red* (resp. *orange*). We also shall call cells of these Θ-bands *red* (resp. *orange*).

Since Θ-bands cannot intersect, there could be at most one i between 1 and $4N-4$ such that t_i contains both red and orange edges. Also it is clear that if t_i contains a red (resp. orange) edge then t_{i+1} (resp. t_{i-1}) contains no orange (resp. red) edges. Therefore either $t_1, \ldots, t_{2N-3}$ contains no red edges or $t_{2N}, \ldots, t_{4N-4}$ contain no orange edges. Without loss of generality we assume that the second possibility holds.

In this case let us erase the red paint from all red edges on the last 4 of the t_j's ($j = 4N-7, 4N-6, 4N-5, 4N-4$). We shall save the last four of Σ_j for the later construction, so we shall not touch the corresponding t_j and y_j by a paint brush. Notice that by our assumption these t_j do not have orange edges. Therefore every Θ-band $\mathcal{R}$ in Ψ_1 which crosses $\mathcal{B}_{4N-7}$, must have one of its ends on a t_ℓ with $\ell < 4N-7$. We paint this end of $\mathcal{R}$ in *red* if it is not red already. Then we call the maximal Θ-band starting on this edge *red* as well.

Among the $2N-7$ numbers $2N, \ldots, 4N-8$ exactly $N-4 \geq 2$ numbers are odd. Notice that if i is odd then y_i starts with a $\mathbf{E}(0)$-edge.

Pick one of the odd numbers $i \in \{2N+1, 2N+3, \ldots, 4N-11\}$. It will be clear later why we do not consider $i = 4N-9$.

Let W' be the word written on y_i. We shall compare $||W'||$ and the length of t_i. In order to do that we establish a correspondence between $\alpha^{\pm 1}$-edges of y_i and edges of t_i. Then we shall prove that only a constant number of $\alpha^{\pm 1}$-edges on y_i can correspond to the same edge in t_i.

Let us paint all colorless Θ-edges of t_i in *green.* All cells of the maximal Θ-bands in Σ_i starting on these edges will be called *green.* Notice that since t_i contains no orange edges, all Θ-edges of t_i are either green or red. Therefore the Θ-band starting on a green edge of t_i ends on p_1. Since by definition of y_i no Θ-band in Σ_i which starts on $\mathcal{B}_i$ can end on $\mathcal{B}_{i+1}$, every transition cell in Σ_i is either red or green.

We shall paint an $\alpha^{\pm 1}$-edge of y_i in *green* if the maximal $\bar{Y}$-band in Σ_i starting on this edge ends on a green transition cell.

Lemma 11.8 *No more than 4 $\bar{Y}$-bands starting on $\alpha^{\pm 1}$-edges of y_i can end on a green cell belonging to the same Θ-band.*

Proof. Indeed, suppose that 5 $\bar{Y}$-bands $\mathcal{Y}_1,...,\mathcal{Y}_5$ starting on α-edges e_1, e_2, e_3, e_4, e_5 end on the contour of the same Θ-band $\mathcal{R}$. We can assume that e_1 precedes e_2 precedes ... e_5 on y_i. Then first three of these edges are to the left of the $\mathbf{X}(0)$-edge of y_i or the last three of these edges are to the right of this $\mathbf{X}(0)$-edge. These two cases are similar so we shall consider only the first case. Let ρ_1, ρ_2, ρ_3 be the green cells on the contour of which the $\bar{Y}$-bands $\mathcal{Y}_1, \mathcal{Y}_2, \mathcal{Y}_3$ end. These cells are transition cells corresponding to the rules of $\mathcal{S}$ involving α. Such transition cells must contain a $\mathbf{X}(0)$-edge on the contour, so ρ_i, ρ_2 and ρ_3 belong to some maximal $\mathbf{X}(0)$-bands $\mathcal{X}_1$, $\mathcal{X}_2$, $\mathcal{X}_3$ respectively (see Fig. 15). Consider the subdiagram Σ' of Σ_i bounded by $\mathcal{Y}_1$, $\mathcal{Y}_3$, $\mathcal{R}$ and y_i. The part of $\mathcal{X}_2$ which is contained in Σ' cannot cross $\mathcal{R}$ (Lemma 7.4). It also cannot cross $\mathcal{Y}_1$ and $\mathcal{Y}_3$ because $\mathcal{Y}_1$ or $\mathcal{Y}_3$ would contain a transition cells and $\bar{Y}$-bands cannot contain transition cells. Therefore $\mathcal{X}_2$ must start or end on y_i between e_1 and e_2.

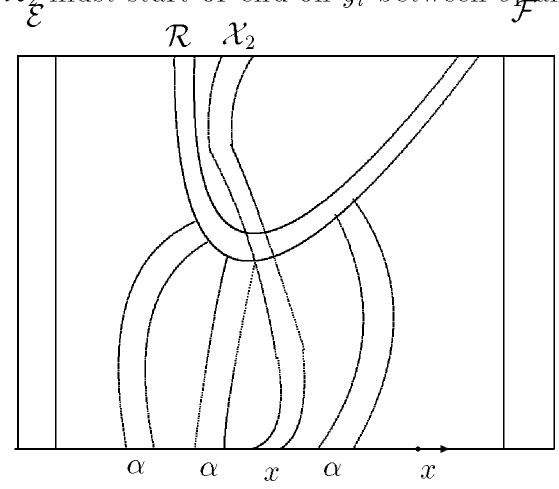

Figure 15.

But this contradicts the fact that there are no $\mathbf{X}(0)$-edges between e_1 and e_3. This contradiction proves our lemma. $\Box$

Notice that for every $x \in \mathbf{X}(0)$ there exists at most one relation in $\mathcal{P}_N(\mathcal{S})$ which does not contain $\mathbf{E}(0)$- or $\mathbf{F}(0)$-letters but contains α.

This relation has one of the forms $x^\tau(\alpha x \alpha^{-1})^{-1} = 1$ or $x^\tau(\alpha^{-1} x \alpha)^{-1} = 1$. Such relations will be called *$x\alpha$-relations.* A cell corresponding to an $x\alpha$-relation will be called an $x\alpha$-cell. The contour of every $x\alpha$-cell has two α-edges with opposite orientation. Let us mark these edges and consider α-*bands* consisting of auxiliary α-cells and these $x\alpha$-cells with common α-edges. Notice that every $\bar{Y}$-band starting on an $\alpha^{\pm 1}$-edge is an α-band, but α-bands are in general longer, they can contain some transition cells ($x\alpha$-cell). An α-band in Σ_i starting on y_i can end either on t_i or on y_i or on a contour of a transition $\mathbf{F}(0)$-cell.

We shall need two general lemmas about α-bands.

Lemma 11.9 *If a $\bar{Y}$-band $\mathcal{Y}$ in a reduced diagram starts on the contour of an x-cell π_1 and ends on the contour of an x-cell π_2 and these two cells belong to the same $\mathbf{X}(0)$-band $\mathcal{B}$ then $\mathcal{B}$ contains an $\mathbf{E}(0)$-cell or an $\mathbf{F}$-cell between π_1 and π_2.*

Proof. We can assume that π_1 is the first cell in $\mathcal{B}$ and π_2 is the last cell in this band. Suppose that $\mathcal{B}$ does not have a $\mathbf{E}(0)$-cell and does not have a $\mathbf{F}(0)$-cell between π_1 and π_2. Then by Proposition 4.1 all $\mathbf{X}(0)$-edges of $\mathcal{B}$ are the same. Therefore all these cells correspond to the same $x\alpha$-relation $x^\tau(\alpha^{\pm 1}x\alpha^{\mp 1})^{-1}$ therefore the labels of $\mathbf{top}(\mathcal{B})$ and $\mathbf{bot}(\mathcal{B})$ have the form $(\tau^a\alpha^b)^\ell$ where $a, b \in \{-1, 1\}$. Therefore all α-edges on $\mathbf{top}(\mathcal{B})$ (resp. $\mathbf{bot}(\mathcal{B})$) have the same orientation. Therefore our $\bar{Y}$-band $\mathcal{Y}$ cannot start and end on $\mathbf{top}(\mathcal{B})$ and cannot start and end on $\mathbf{bot}(\mathcal{B})$. But it also cannot start (end) on $\mathbf{top}(\mathcal{B})$ and end (start) on $\mathbf{bot}(\mathcal{B})$ because otherwise the maximal $\mathbf{X}(0)$-band containing $\mathcal{B}$ would intersect the $\bar{Y}$-band $\mathcal{Y}$ and a $\bar{Y}$-band cannot have transition cells.□

Lemma 11.10 *A reduced van Kampen diagram over $\mathcal{P}_N(\mathcal{S})$ without hubs cannot contain α-annuli.*

Proof. Indeed, let $\mathcal{A}$ be an α-annulus. If $\mathcal{A}$ does not have $\mathbf{X}(0)$-cells then it is a $\bar{Y}$-annulus which is ruled out by Lemma 7.9. So $\mathcal{A}$ must contain an $\mathbf{X}(0)$-cell. Thus it has at least 2 $\mathbf{X}(0)$-edges on its inner contour. Therefore a $\mathbf{X}(0)$-band $\mathcal{B}$ intersects the annulus $\mathcal{A}$ twice. Then $\mathcal{B}$ cuts $\mathcal{A}$ into two parts, $\mathcal{A}_1$ and $\mathcal{A}_2$. We can assume that $\mathcal{A}_1$ does not contain $\mathbf{X}(0)$-cells. Then $\mathcal{A}_1$ is a $\bar{Y}$-band starting and ending on the contour of an $\mathbf{X}(0)$-band $\mathcal{B}$. By Lemma 11.9 this band must contain an $\mathbf{E}(0)$ or an $\mathbf{F}(0)$-cell π between the two intersections with $\mathcal{A}$. Since $\mathcal{A}$ cannot have $\mathbf{E}(0)$-cells or $\mathbf{F}(0)$-cells, π must be inside the subdiagram bounded by the median of $\mathcal{A}$. But then the maximal $\mathbf{E}(0)$ or $\mathbf{F}(0)$-band containing π must intersect with $\mathcal{A}$. This is a contradiction since $\mathcal{A}$ cannot contain $\mathbf{E}(0)$ and $\mathbf{F}(0)$-cells. □.

Now we can continue painting edges of y_i and t_i.

If an α-band starts on a green edge of y_i and ends on y_i^{-1} then we paint the inverse of the end edge (it belongs to y_i) green also.

Lemma 11.11 *The number of green edges on y_i cannot exceed 8 times the number of green edges on t_i.*

Proof. Indeed, by the definition of green edges, at least half of the green edges e on y_i have the property that the $\bar{Y}$-band $\mathcal{Y}$ starting on e ends on a green Θ-cell ρ. If e is such an edge then we associate with e the green edge on t_i which is the start or the end edge of the maximal Θ-band containing ρ (whichever of these two edges belongs to t_i; if both start and end edges of this Θ-band are on t_i then we take the leftmost edge). By Lemma 11.8 at most 4 green edges of y_i are associated with the same green edge on t_i. Therefore the total number of green edges on y_i does not exceed 8 times the number of green edges of t_i. $\Box$

Now we need more colors.

If an α-band starting on y_i ends on t_i then we paint the start and the end edges of this band in yellow. The following lemma is obvious.

Lemma 11.12 *The number of yellow edges on y_i coincides with the number of yellow edges on t_i.*

Let us denote the maximal $\mathbf{E}(0)$-band starting on the $\mathbf{E}(0)$-edge of y_i by $\mathcal{E}$, the maximal $\mathbf{F}(0)$-band starting on y_i by $\mathcal{F}$ and the maximal $\mathbf{X}(0)$-band starting on y_i by $\mathcal{X}$.

If an α-band $\mathcal{A}$ starting on a non-green and non-yellow edge e of y_i ends on the contour of a red $\mathbf{F}(0)$-cell such that the maximal $\mathbf{F}(0)$-bands $\mathcal{F}_e^{\pm 1}$ containing this red cell are not $\mathcal{F}^{\pm 1}$ then both start and end edge of $\mathcal{F}_e$ belong to t_i (an $\mathbf{F}(0)$-band cannot cross a κ-band). We paint e and the start and end edges of $\mathcal{F}_e$ in *blue*. See Fig. 16.

Lemma 11.13 *No more than two maximal α-bands starting on blue edges of y_i can end on the contour of the same $\mathbf{F}(0)$-band.*

Proof. If there are three such maximal α-bands then at least two of them have start edges on the same side of the $\mathbf{X}(0)$-edge of y_i. Let e_1 and e_2 be blue edges of $(y_i)^{\pm 1}$ which are on the same side of the $\mathbf{X}(0)$-edge of $(y_i)^{\pm 1}$, $e_1^{\pm 1}$ preceding $e_2^{\pm 1}$ on y_i, let $\mathcal{A}_1$ and $\mathcal{A}_2$ be the maximal α-bands starting on e_1 and e_2 respectively. Then the edges e_1 and e_2 either both belong to y_i or both belong to $(y_i)^{-1}$ (if these edges were on opposite sides of the $\mathbf{X}(0)$-edge of y_i, one of them could belong to y_i and the other one could belong to $(y_i)^{-1}$).

Suppose that $\mathcal{A}_i$, $i = 1, 2$, ends on the contour of a red cell ρ_i. Let $\mathcal{R}_1$ and $\mathcal{R}_2$ be the red Θ-bands containing ρ_1 and ρ_2. Let $\mathcal{F}'$ be one of the two mutually inverse maximal $\mathbf{F}(0)$-bands containing ρ_1 and ρ_2.

Notice that the $\mathbf{F}(0)$-band $\mathcal{F}'$ cannot intersect $\mathcal{R}_1$ or $\mathcal{R}_2$ twice by Lemma 7.4. Therefore the (red) start edges of $\mathcal{R}_1$ and $\mathcal{R}_2$ are between the start and the end edges of $\mathcal{F}'$. Notice also that between the α-edge and the $\mathbf{F}(0)$-edge on the top or bottom of a $\mathbf{F}(0)$-cell considered as a Θ-band, there always exists a $\mathbf{X}(0)$-edge. This implies that if one of the maximal $\mathbf{X}(0)$-bands containing ρ_1 or ρ_2 is inside the subdiagram Σ' bounded by $\mathcal{F}'$ and t_i then the α-band $\mathcal{A}_1$ must intersect $\mathcal{F}$ which is impossible. Therefore the maximal $\mathbf{X}(0)$-bands containing ρ_1 (resp. ρ_2) must be outside Σ' (recall that an $\mathbf{X}(0)$-band and an $\mathbf{F}(0)$-band cannot cross).

Since the Θ-bands $\mathcal{R}_1$ and $\mathcal{R}_2$ cross $\partial(\mathcal{B}_{4N-7})^{\pm 1}$, and since $i < 4N - 10$, $\mathcal{R}_1$ and $\mathcal{R}_2$ cross both κ-bands $\mathcal{B}_{4N-9}$ and $\mathcal{B}_{4N-8}$ in Σ_{4N-9}. It is clear that $\mathcal{R}_2$ is higher than $\mathcal{R}_1$ (that is the intersection of $\mathcal{R}_1$ $\mathcal{B}_{i+1}$ has bigger number on $\mathcal{B}_{i+1}$ than the intersection of $\mathcal{R}_2$ and $\mathcal{B}_{i+1}$) .

The paths $\mathbf{top}(\mathcal{R}_i)$ and $\mathbf{bot}(\mathcal{R}_i)$ start on t_i and cross $\mathcal{B}_{i+1}$. On the boundary of the cell ρ_j, $j = 1, 2$ we can read either the word

$$(x(0, \tau_j, 4)F(0, \tau_j, 4))^\theta (x(0, \tau_j, \alpha)F(0, \tau_j, \alpha)\alpha^{-1})^{-1}$$

or the inverse of this word. In the first case we shall call ρ_j *positive*, in the other case – *negative*.

Since Σ' does not contain the $\mathbf{X}(0)$-edge of ρ_j, either both the edge labelled by $F(0,\tau_j,4)$, and the edge labelled by $x(0,\tau_j,4)$ belong to the path $\mathbf{top}(\mathcal{R}_j)^{-1}$ or both of these edges belong to the path $\mathbf{bot}(\mathcal{R}_j)^{-1}$, so they are not oriented toward $\mathcal{B}_{i+1}$. Similarly the edge of ρ_j labelled by $F(0,\tau_j,\alpha)$, and the edge labelled by $x(0,\tau_j,\alpha)$ belong both to the path $\mathbf{bot}(\mathcal{R}_j)^{-1}$ or both to the path $\mathbf{top}(\mathcal{R}_j)^{-1}$.

Let e_j' be the α-edge of ρ_j. Then e_j' is the end edge of the α-band $\mathcal{A}_j$. In every cell of $\mathcal{A}_j$ two α-edges form an opposing pair. Therefore e_j' and e_j form an opposing pair of edges on the boundary of $\mathcal{A}_j$. Since e_1 and e_2 either both point toward $\mathcal{B}_{i+1}$ or both point in the opposite direction, the edges e_1' and e_2' must also have the same orientation meaning that if we trace the boundary of ρ_1 and ρ_2 clockwise then we either pass through e_1' and e_2' or we pass through $(e_1')^{-1}$ and $(e_2')^{-1}$.

This implies that either both ρ_1 and ρ_2 are positive or both are negative. Therefore the common Θ_+-edges of ρ_j with $\mathcal{B}_{4N-9}$ either both point toward y_{4N-9} or both point in the opposite direction.

The subdiagram Γ of Σ_{4N-9} bounded by y_{4N-9} on the bottom, $\mathcal{R}_2$ on the top, $\mathcal{B}_{4N-9}$ on the left and $\mathcal{B}_{4N-8}$ of the right is a sector. By Proposition 9.1 we can assume that Γ is a computational sector. Let $C = W_1, ..., W_m$ be the corresponding computation connecting $W_1 = \mathrm{Lab}(\mathbf{top}(\mathcal{R}_2)$ and $\mathrm{Lab}(y_{4N-9})$. Then $\mathrm{Lab}(\mathbf{bot}(\mathcal{R}_1))$ is one of the words W_n. Since there exists a computation of $\mathcal{S}$ connecting W_m and W_0, there also exist computations connecting W_1 and W_n with W_0. The rules used in the transitions $W_1 \to W_2$ and $W_{n-1} \to W_n$ are $R_{4,\alpha}(\tau_1)^{\pm 1}$ and $R_{4,\alpha}(\tau_2)^{\pm 1}$. The fact that we proved in the previous paragraph shows that either both of these rules are positive or both are negative. Therefore part 5 of Proposition 4.1 applies either to the computation $W_1, ..., W_n$ or to the inverse computation. In any case, in this computation, there exists a transition $W_\ell \to W_{\ell+1}$ such that the rule τ applied in this transition contains a word Ex as one of its left sides where $E \in \mathbf{E}(0)$, $x \in \mathbf{X}(0)$. Let $\bar{\mathcal{R}}_3$ be the maximal Θ-band corresponding to this transition in the computational sector Γ. Let $\mathcal{R}_3$ be the maximal Θ-band in Ψ_1 containing $\bar{\mathcal{R}}_3$.

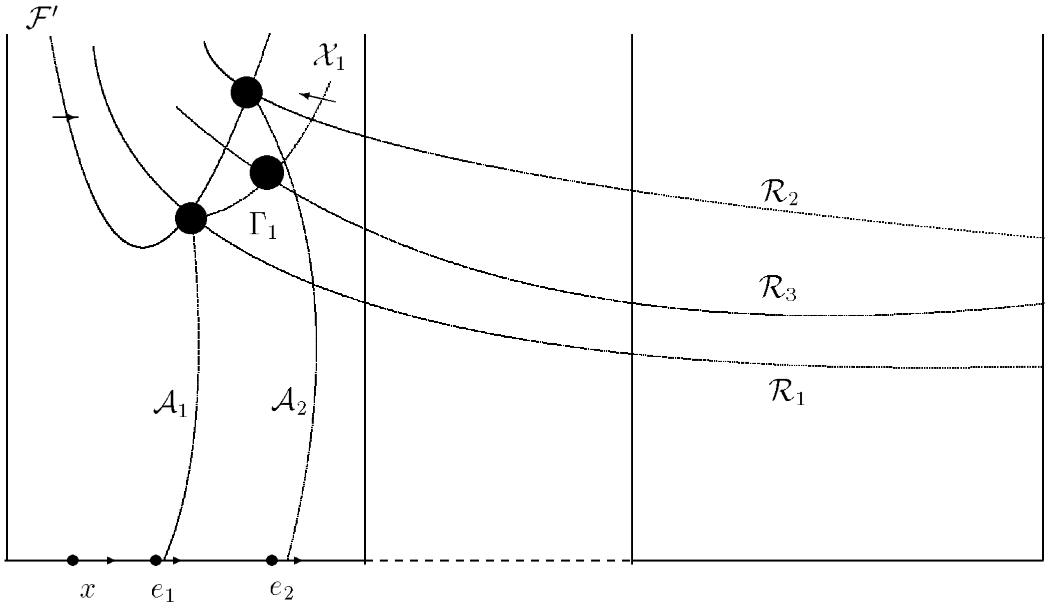

Figure 16.

Let Γ_1 be the subdiagram of Ψ_1 bounded by $\mathcal{B}_{i+1}$, $\mathcal{R}_1$, $\mathcal{R}_2$, $\mathcal{F}'$. Let $\mathcal{X}_1$ be the maximal $\mathbf{X}(0)$-band in Γ_1 starting on the boundary of ρ_1. Then $\mathcal{X}_1$ cannot have two common cells with $\mathcal{R}_1$, it cannot cross $\mathcal{B}_{i+1}$ and it cannot cross $\mathcal{F}'$ (it can only touch $\mathcal{F}'$). Therefore $\mathcal{X}_1$ crosses $\mathcal{R}_2$. Since $\mathcal{R}_3$ is between $\mathcal{R}_1$ and $\mathcal{R}_2$, it must cross $\mathcal{X}_1$. Let ρ be the intersection cell. Then ρ corresponds to the relation containing both an $\mathbf{X}(0)$-letter and τ. By definition of $\mathcal{P}_N(\mathcal{S})$ there is only one (modulo taking inverses and cyclic shifts) relation which contains both an $\mathbf{X}(0)$-letter and τ. This relation or its inverse is written on the boundary of the cell in $\mathcal{R}_3$ which contains an $\mathbf{X}(0)^{\pm 1}$-edge. By the choice of τ, this relation contains an $(\mathbf{E}(0))$-letter. Therefore the boundary of ρ contains an $\mathbf{E}(0)$-edge. Thus Γ_1 contains an $\mathbf{E}(0)$-edge. Let $\mathcal{E}_1$ be the maximal $\mathbf{E}(0)$-band in Σ_i containing this edge. The $\mathbf{E}(0)$-band $\mathcal{E}_1$ cannot cross $\mathcal{F}'$ or $\mathcal{B}_{i+1}$. By Lemma 7.2, $\mathcal{E}_1$ is not an annulus. Therefore $\mathcal{E}_1$ crosses $\mathcal{R}_1$ between ρ_1 and $\mathcal{B}_{i+1}$. Since the $\mathbf{E}(0)$-band $\mathcal{E}_1$ cannot cross $\mathcal{A}_1$, either the start or the end edge of $\mathcal{E}_1$ must belong to $y_i^{\pm 1}$. This edge must be between e_1 and the $(\mathbf{F}(0))^{\pm 1}$-edge of $(y_i)^{\pm 1}$. But $(y_i)^{\pm 1}$ does not contain $(\mathbf{E}(0))^{\pm 1}$-edges between the $\alpha^{\pm 1}$-edges and the $(\mathbf{F}(0))^{\pm 1}$-edge. This contradiction completes the proof of our lemma. $\Box$.

This lemma immediately implies the following result.

Lemma 11.14 *The number of blue edges on y_i does not exceed the number of blue edges on t_i.*

We continue painting the edges of y_i. We shall paint an $\alpha^{\pm 1}$-edge e of y_i in *pink* if the α-band starting on this edge ends on the contour of a green Θ-cell.

Lemma 11.15 *There are no more than two α-bands starting on $(y_i)^{\pm 1}$ and ending on the boundary of the same green Θ-band.*

Proof. The proof is similar to the proof of Lemma 11.8. Suppose that there are three such α-bands $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$ and the start edge of $\mathcal{A}_1$ precedes the start edge of $\mathcal{A}_2$ which precedes the start edge of $\mathcal{A}_3$. Let ρ_1, ρ_2 and ρ_3 be the green cells containing the end edges of $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{A}_3$ respectively and belonging to the same Θ-band $\mathcal{R}$. Then the maximal $\mathbf{F}(0)$-band $\mathcal{F}_2$ containing ρ_2 cannot cross $\mathcal{R}$ again and cannot cross $\mathcal{A}_1$ and $\mathcal{A}_3$. Thus $\mathcal{F}_1$ must start or end on $(y_i)^{\pm 1}$. This contradicts the fact that y_i cannot contain an $(\mathbf{F}(0))^{\pm 1}$-edge between two $\alpha^{\pm 1}$-edges. $\square$

This lemma implies the following fact.

Lemma 11.16 *The number of pink edges on y_i does not exceed twice the number of green edges on t_i.*

Our next goal will be to prove that every $\alpha^{\pm 1}$-edge on y_i is either green, yellow, blue or pink.

Let us suppose that it is not so. Then there exists a colorless edge $e \in (y_i)^{\pm 1}$ labelled by α such that the maximal α-band starting on e ends either on an edge e' of $(y_i)^{-1}$ or on the contour of a $\mathbf{F}(0)$-cell which belongs to the $\mathbf{F}(0)$-band $\mathcal{F}$ and to a red Θ-band. In the first case e' is colorless (it could be only green but then e would also be green and we assumed that e is colorless). In this case we paint both e and $(e')^{-1}$ in *brown*. In the second case we paint e in *black*.

Notice that if there are brown or black edges on y_i then t_i contains red edges. Therefore there exists a Θ-band in Ψ_1 which starts on t_i and crosses $\mathcal{B}_{4N-7}$.

Lemma 11.17 *y_i does not contain brown edges.*

Proof. Suppose that y_i contains a brown edge e. Let e' be the end edge of the maximal α-band $\mathcal{A}$ in Σ_i starting on e. As we know, e' is also brown. Since e belongs to y_i, e' belongs to y_i^{-1} and the labels of e and e' are the same. Therefore these edges cannot be on the same side of the $\mathbf{X}(0)$-edge of y_i (since the label of y_i is a reduced word). Thus one of these edges is between the $\mathbf{E}(0)$-edge and the $\mathbf{X}(0)$-edge and the other one is between the $\mathbf{X}(0)$-edge and the $\mathbf{F}(0)$-edge. Suppose that e is between the $\mathbf{E}(0)$-edge and the $\mathbf{X}(0)$-edge. The case when e' is between the $\mathbf{E}(0)$-edge and the $\mathbf{X}(0)$-edge is completely similar (although not identical because we have the additional assumption that $e \in y_i$).

Since $e \in y_i$ and the label of e is α, the part of the word $\mathrm{Lab}(y_i)$ between the $\mathbf{E}(0)$-letter and the $\mathbf{F}(0)$-letter has the form $\alpha^m x \alpha^n$ where $m > 0, n < 0$.

Since e is not a green or yellow edge, the $\bar{Y}$-band starting on e ends on a red Θ-cell. Let $\mathcal{R}'$ be the maximal red Θ-band which contains this cell. Since e is to the left of the $\mathbf{X}(0)$-band $\mathcal{X}$, $\mathcal{R}'$ intersects $\mathcal{X}$.

Let ρ be the first cell in $\mathcal{X}$ (its contour has a common path with y_i^{-1}). Let $\mathcal{R}$ be the maximal Θ-band in Ψ_1 containing ρ. Since $\mathcal{R}$ cannot intersect $\mathcal{R}'$, it must cross $\mathbf{bot}(\mathcal{B}_{4N-7})$. So $\mathcal{R}'$ is a red Θ-band (see Fig. 17).

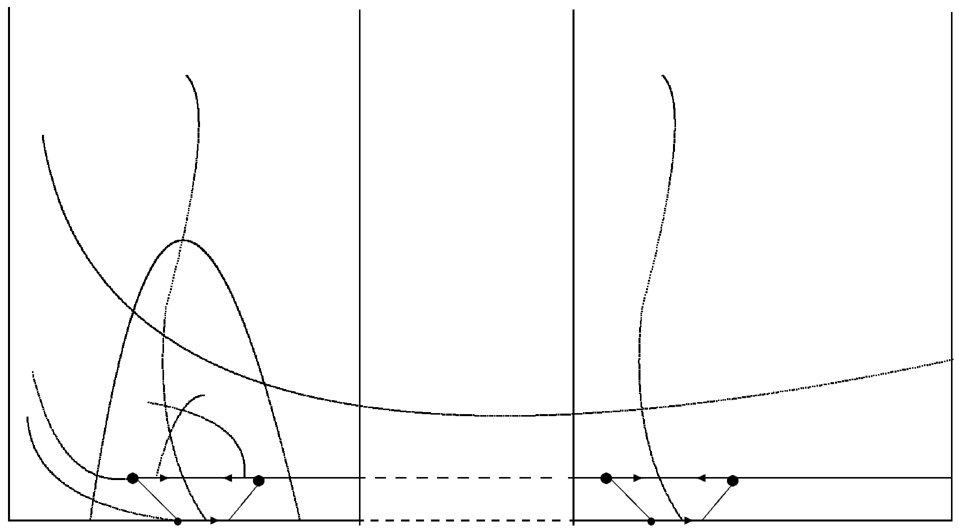

Figure 17.

Let Σ' be the diagram bounded by $\mathcal{R}$, $\mathcal{B}_{4N-9}$, $\mathcal{B}_{4N-8}$, y_{4N-9}. Then Σ' is a sector which we can assume to be a computational sector. This sector has only two mutually inverse maximal Θ-bands (one of which is a part of $\mathcal{R}$). Indeed, if there were a Θ-band $\mathcal{R}''$ in Σ' below $\mathcal{R}$ then the maximal Θ-band in Ψ_1 containing $\mathcal{R}''$ would intersect $\mathcal{X}$ and the intersection cell would be lower than ρ. Thus the computation corresponding to the sector Σ' consists of one transition $W_1 \to W_2$ where $W_1 = \mathrm{Lab}(y_{4N-9})$. Recall that the α-part of $\mathrm{Lab}(y_{4N-9})$ is equal to $\alpha^m x \alpha^n$ where $n \neq 0$ and $m \neq 0$. Hence the only rules applicable to W_1 are rules of the form $x^\tau(\alpha^\epsilon x \alpha^{-\epsilon})^{-1}$. If $\epsilon = -1$ then $|W_2| < |W_1|$, a contradiction with property P7. Thus the boundary of the cell in Σ' which has an $\mathbf{X}(0)$-edge has label $\tau\alpha x \alpha^{-1}\tau^{-1}x^{-1}$. Denote this cell by ρ''. Since the cells ρ and ρ'' belong to the same Θ-band and their $\mathbf{X}(0)$-edges have the same orientation on s_1, on the boundary of ρ one can read the same word. Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be the maximal α-bands in Σ_i which start on the two α-edges e_1 and e_2 of the contour of ρ where e_1 belongs to the subdiagram bounded by $\mathcal{E}$, $\mathcal{X}$, y_i and t_i and e_2 belongs to the complement of this subdiagram in Σ_i. These α-bands cannot intersect the α-band $\mathcal{A}$. Therefore they must end on $y_i^{\pm 1}$ (α-annuli are ruled out by Lemma 11.10). The α-band α_1 cannot end to the left of the $\mathbf{X}(0)$-edge of y_i because the start and the end edges of an α-band must form an opposing pair. Therefore the end edge of $\mathcal{A}_1$ is to the right of the $\mathbf{X}(0)$-edge of y_i. Similarly the end edge of $\mathcal{A}_2$ must be to the left of this $\mathbf{X}(0)$-edge. This implies that $\mathcal{A}_1$ and $\mathcal{A}_2$ must intersect, a contradiction. □

Lemma 11.18 *y_i does not contain black edges.*

Proof. Let e be a black α-edge on $y_i^{\pm 1}$. Let $\mathcal{A}$ be the maximal α-band starting on e, and let ρ be the cell from $\mathcal{F}$ on whose boundary $\mathcal{A}$ ends. Let $\mathcal{R}$ be the maximal red Θ-band containing ρ (see Fig. 18).

The subdiagram Γ bounded by $\mathcal{B}_{4N-9}$, $\mathcal{B}_{4N-8}$, $\mathcal{R}$ and y_{4N-9} is a sector and by Proposition 9.1 can be converted into a computational sector without changing the boundary or increasing the area. So we can assume that Γ is a computational sector. Let $C = (W_1, ..., W_\ell)$ be the corresponding computation where W_1 (resp. W_2) is the label of the part of $\mathbf{top}(\mathcal{R})$ (resp. $\mathbf{bot}(\mathcal{R})$) between $\mathcal{B}_{4N-9}$ and $\mathcal{B}_{4N-8}$, $W_\ell = \mathrm{Lab}(y_{4N-9}) = \mathrm{Lab}(y_i) = W$. The label θ of the Θ-edges of $\mathcal{R}$ is an S-rule which involves α and an $\mathbf{F}(0)$-letter. Therefore this rule is $R_{4,\alpha}(\tau)$ for some τ. Since there exists a computation of $\mathcal{S}$ connecting W_ℓ and W_0 (this is the computation corresponding to the computational disc Π), there exists a reduced computation connecting W_0 and W_1. By Proposition 4.1 either W_1 or W_2 is a positive word and the degree of α in W_1 and W_2 is positive.

Suppose that W_ℓ contains α^{-1}. Let W_m, $m < \ell$, be the last word in the computation C which does not contain α^{-1}. Then the α-part of W_{m+1} has the form $\alpha^a x \alpha^b$ where $x \in \mathbf{X}(0)$, $ab < 0$. Therefore only $x\alpha$-rules are applicable to W_{m+1}. Since the computation C is semiproper, all words in the computation $W_{m+1}, ..., W_\ell$ contain α^{-1}, so in this computation only $x\alpha$-rules can apply and each of them inserts a new α^{-1}. Since $x\alpha$-rules do not affect non-α parts of admissible words, the length of W_ℓ is greater than the length of W_{m+1} provided $m+1 < \ell$. This contradicts Property P7. Thus $m+1 = \ell$. The only rules which can insert α^{-1} are $R_{4,\alpha}(\tau)$ and $x\alpha$-rules. Thus one of these rules is applied in the transition $W_m \to W_\ell$. But if $R_{4,\alpha}(\tau)$ applies in this transition then the degree of α in W_m should be 0. Since W_m is normal and positive, it must contain no $\bar{Y}$-letters. Then $|W_{m+1}| = |W_m| + 4$ which contradicts Property P7.

If an $x\alpha$-rule is applied in the transition $W_m \to W_{m+1}$ then $|W_{m+1}| = |W_m| + 2$ and we again get a contradiction.

Thus the α-part of W_ℓ does not contain α^{-1}, so all $\alpha^{\pm 1}$-edges of y_{4N-9} have label α. Since the labels of y_i and y_{4N-9} are the same, all $\alpha^{\pm 1}$-edges of y_i have label α.

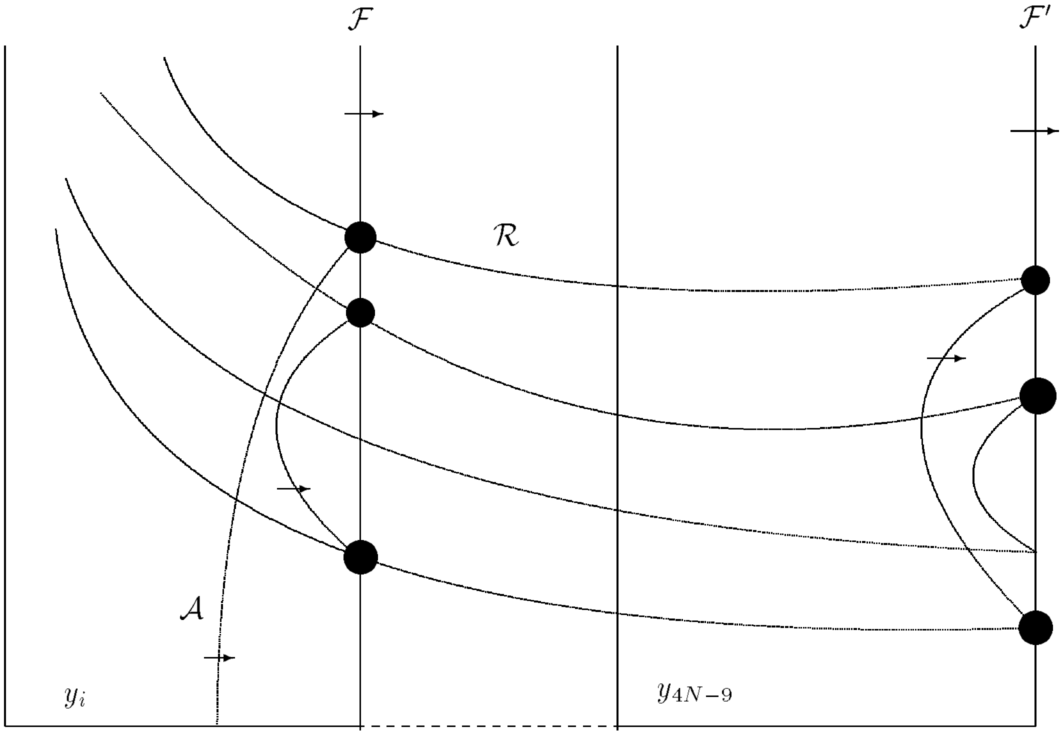

Figure 18.

Since e has label α, it must belong to y_i. The start and end edges of every α-band form an opposing pair on the contour of this band. Therefore the end edge of $\mathcal{A}$ must belong to the inverse of the boundary of the cell ρ. Therefore on the boundary of ρ, one can read $F^{-1}x^{-1}\gamma^{-1}\alpha^{-1}x'F'\gamma$ where $\gamma \in \Theta_+$, $x, x' \in \mathbf{X}(0)$, $F, F' \in \mathbf{F}(0)$. Therefore the same word can be read on the boundary of the intersection ρ' of $\mathcal{R}$ and the $\mathbf{F}(0)$-band $\mathcal{F}'$ of Γ starting on y_{4N-9}. Hence the rule applied in the transition $W_1 \to W_2$ in the computation C is $R_{4,\alpha}(\tau)^{-1}$ for some τ. Hence the length of W_2 is greater than the length of W_1 (Proposition 4.1).

Let $\mathcal{R}'_1, ..., \mathcal{R}'_{\ell-1}$ be the maximal Θ-bands of Γ starting on the contour of $\mathcal{B}_{4N-9}$ counted from top to bottom (so $y_{4N-9} = \mathbf{bot}(\mathcal{R}_{\ell-1})$) and let $\mathcal{R}_1, \mathcal{R}_2, ..., \mathcal{R}_{\ell-1}$ be the red Θ-bands of Ψ_1 containing $\mathcal{R}'_1, ..., \mathcal{R}'_{\ell-1}$ (so $\mathcal{R} = \mathcal{R}_1$). Then W_2 is written on the bottom of $\mathcal{R}'_1$. Let e'_1 be the last α-edge of $\mathbf{bot}(\mathcal{R}'_1)$. Let $\mathcal{A}'_1$ be the maximal α-band in Γ starting on e'_1. If $\mathcal{A}'_1$ ends on y_{4N-9} then the degree of α in W_ℓ is greater than the degree of α in W_1. Since W_1 is a positive word (by Proposition 4.1), W_ℓ is longer than W_1 which contradicts Property P7.

The α-band $\mathcal{A}'_1$ cannot end on the contour of $\mathcal{R}_1$ because then W_1 would contain α^{-1}. Therefore the only possibility is that $\mathcal{A}'_1$ ends on the contour of the $\mathbf{F}(0)$-band $\mathcal{F}'$. Let the $\mathbf{F}(0)$-cell ρ'_1 on the contour of which this band ends belong to the Θ-band $\mathcal{R}_{j_1}$. Considering the orientation of α-edges of $\mathbf{bot}(\mathcal{R}_1)$ and the orientation of ρ'_1 we conclude that the rule applied in the transition $W_{j_1} \to W_{j_1+1}$ in the computation C is $R_{4,\alpha}(\tau_1)$ for some τ_1. Let ρ_1 be the intersection of $\mathcal{R}_{j_1}$ and $\mathcal{F}$. Then the maximal α-band $\mathcal{A}_1$ starting on the α-edge of the contour of ρ_1 cannot cross $\mathcal{A}$ and cannot end on $y_i^{\pm 1}$ (because all

$\alpha^{\pm 1}$-edges of y_i have label α). Therefore $\mathcal{A}_1$ must end on the contour of an $\mathbf{F}(0)$-cell ρ_2 from $\mathcal{F}$. This cell must belong to a red Θ-band $\mathcal{R}_{j_2}$ for some j_2 between 1 and j_1. Let ρ_2' be the intersection of $\mathcal{R}_{j_2}$ with $\mathcal{F}'$. Then the α-band $\mathcal{A}_2'$ starting on the α-edge of the contour of ρ_2' ends on the contour of some cell ρ_3' of $\mathcal{F}$ (it cannot cross $\mathcal{A}_1'$), and the red Θ-band $\mathcal{R}_{j_3}$ containing this cell intersects $\mathcal{F}$ at some cell ρ_3. Continuing in this manner we shall construct a sequence of different cells $\rho_1, \rho_2, \rho_3, \dots$ of $\mathcal{F}$. This process obviously cannot stop which contradicts the fact that $\mathcal{F}$ contains only finitely many cells (see Fig. 18). This contradiction completes the proof of the lemma. $\Box$

Thus we have proved that every $\alpha^{\pm 1}$ edge of y_i is either green or yellow or blue or pink. This allows us to prove the following statement.

Let n_i be the number of non-red edges of t_i. Recall that W denotes the label of y_i.

Lemma 11.19 $||W|| = ||\mathrm{Lab}(y_i)|| \leq 40n_i + 15k + 6 < 61kn_i$ *(where k is the number of tapes of the machine M).*

Proof. Indeed, by Lemmas 11.11, 11.12, 11.14, 11.16, the number of green and blue edges on y_i does not exceed $8+2 = 10$ times the number of green edges of t_i, the number of yellow edges of y_i does not exceed the number of yellow edges of t_i and the number of pink edges of y_i does not exceed the number of pink edges on t_i. Therefore the degree m_i of α in $W = \mathrm{Lab}(y_i)$ does not exceed $10n_i$. Since W is a normal word, $||W|| = 4m_i + 15k + 6$. This implies the inequalities of the lemma. $\Box$

We shall also need to estimate the length of W.

Lemma 11.20 $|W| \leq (30k + 12)|p_1|$

Proof. Indeed, every $\bar{Y}$-band in Ψ_1 starting on y_i (i between 1 and $4N-4$) ends either on p_1 or on the contour of a maximal Θ-band in Ψ. Since every maximal Θ-band either starts or ends on p_1, it is enough to show that the number of maximal $\bar{Y}$-bands in Ψ_1 starting on y_i and ending on the contour of the same Θ-band does not exceed $30k + 12$. The proof is similar to the proof of Lemma 11.8. Suppose that there exist $30k + 13$ $\bar{Y}$-bands starting on y_i and ending on the contour of the same Θ-band $\mathcal{R}$. Since the number of Q-edges on y_i is $15k + 6$, there exist three $\bar{Y}$-bands $\mathcal{Y}_1, \mathcal{Y}_2, \mathcal{Y}_3$ ending on $\mathcal{R}$ and starting on edges e_1, e_2, e_3 of y_i such that e_1 precedes e_2 which precedes e_3, and there are no Q-edges between e_1 and e_3. Then the maximal Q-band containing the intersection of $\mathcal{Y}_2$ and $\mathcal{R}$ must end or start on y_i between e_1 and e_3 because it cannot cross $\mathcal{R}$ twice and cannot cross $\mathcal{Y}_1$ and $\mathcal{Y}_3$. This contradicts the fact that there are no Q-edges between e_1 and e_3 on s_1^{-1}. $\Box$

Now given our decomposition (Ψ_1, Ψ_2, Π) of Δ we define a new decomposition (Γ, E, Π) in the following way. Let Σ' be inverse of the union of subdiagrams Σ_j where $j = 4N-7, 4N-6, 4N-5, 4N-4$ with the κ-bands $\mathcal{B}_{4N-7}$ and $\mathcal{B}_{4N-3}$ removed. Then

$$\partial(\Sigma') = u_1^{-1} q' u_2 (p')^{-1}$$

where $u_1 = \mathbf{bot}(\mathcal{B}_{4N-7})$, $p' \subset p_1$, $u_2 = \mathbf{top}(\mathcal{B}_{4N-3})$, $q' \subset s_1^{-1}$. Notice that $\mathrm{Lab}(u_2) = \mathrm{Lab}(\mathbf{bot}(\mathcal{B}_{4N-3}))$ and $\mathrm{Lab}(q')$ is equal to $\mathrm{Lab}(s_2^{-1})$ if we change indices of the labels of

κ-edges. Let us make this change of indices in the whole Σ' and denote the resulting diagram by Σ.

Then we can glue Σ and Ψ_1 along $\mathbf{bot}(\mathcal{B}_{4N-3})$. The resulting diagram will be denoted by Γ. We also can glue Σ^{-1} and Ψ_2 along $s_2^{-1}\mathbf{bot}(\mathcal{B}_{4N-3})$. The resulting diagram will be denoted by E.

Notice that

$$\partial(\Gamma) = \mathbf{bot}(\mathcal{B}_1)p_1 qus's_1$$

where $|q| = \sum_{j=4N-7}^{4N-3} |t_i| + 3 < |p_1|$, $\mathrm{Lab}(u) = \mathrm{Lab}(\mathbf{bot}(\mathcal{B}_{4N-7}))^{-1}$, $\mathrm{Lab}(s') = \mathrm{Lab}(s_2^{-1})$;

$$\partial(E) = \mathbf{bot}(\mathcal{B}_1)^{-1}u^{-1}q^{-1}p_2.$$

It is clear that by gluing E, Γ and Π we can get a diagram with the same boundary label as Δ. It is also clear that the number of hubs in E is $h(\Delta)-1$ and Γ does not have hubs.

We shall call the triple (Γ, E, Π) a decomposition of type 2 of Δ. We can now estimate the perimeters of Γ, E, Π.

Lemma 11.21 *Let the boundary label of the computational disc Π be $K(W)$ for some word W. Then there exist positive constants ϵ_1 and ϵ_2 such that*

1. $n(\Gamma) \leq \epsilon_1 n(\Delta)$ *and*
2. $n(E) + \epsilon_2 ||W|| \leq n(\Delta)$.

Proof. We have that

$$n(\Gamma) = |\mathbf{top}(\mathcal{B}_1)| + |s_1| + |s_2| + |p_1| + |q| + |u|,$$

$$n(E) = |\mathbf{top}(\mathcal{B}_1) + |u| + |q| + |p_2|.$$

The number $|\mathbf{top}(\mathcal{B}_1)|$ is equal to the number of orange edges on p_1.

The path u consists of Θ-edges (recall that $\mathrm{Lab}(u) = \mathrm{Lab}(\mathbf{top}(\mathcal{B}_{4N-7})))$. No maximal Θ-band $\mathcal{R}$ in Γ starting on u can end on t_j where $j \geq 4N-7$: this follows from the fact that $\Gamma - \Psi_1$ is symmetric to the union of Σ_j, $j = 4N-7, 4N-6, 4N-5, 4N-4$ (with $\mathcal{B}_{4N-3}$ removed). Thus it ends either on t_j for some $j < 4N-7$ or on q. In the first case the end of $\mathcal{R}$ is a red edge. In the second case let e be the end edge of $\mathcal{R}$. Since q with κ-letters removed is symmetric to the union of the t_j's, $j = 4N-7, ..., 4N-4$, there exists a symmetric edge e' on one of these t_j. Then the maximal Θ-band $\mathcal{R}'$ starting on e' must cross $\mathcal{B}_{4N-7}$, therefore it ends on a red edge of one of the t_j's, $j < 4N-7$. The end edge of $\mathcal{R}'$ is red. Thus we have a one-to-one correspondence between edges of u and some red edges on p_1. Therefore the length of u does not exceed (in fact one can prove that it equals) the number of red edges of p_1.

The length of $s_1 s_2^{-1}$ is $4N|W| + 4N$. Therefore by Lemma 11.20

$$|s_1| + |s_2| \leq 4N(30k+12)|p_1| + 4N < 4N(30k+13)|p_1|.$$

Thus

$$n(\Gamma) \leq |p_1| + 4N(30k+13)|p_1| + 2|p_1| < \epsilon_1 n(\Delta)$$

for some ϵ_1.

Notice that $n(\Delta) - n(E) = |p_1| - |\mathbf{top}(\mathcal{B}_1)| - |u| - |q|$. Since the $|\mathbf{top}(\mathcal{B}_1)|$ is equal to the number of orange edges on p_1, $|u|$ does not exceed the number of red edges on p_1, $|p_1| - |\mathbf{top}(\mathcal{B}_1)| - |u|$ exceeds the number of green, yellow, blue and pink edges of p_1 plus the number of colorless edges. By construction $|q|$ is equal to the sum of the lengths of the t_j's, $j = 4N-7, 4N-6, 4N-5, 4N-4$ plus 3 (the κ-edges). Notice that the edges of these t_j are colorless (because we removed the red paint from these t_j and picked $i < 4N-7$ when we starting to paint t_i in different colors). Therefore $n(\Delta) - n(E) = |p_1| - |\mathbf{top}(\mathcal{B}_1)| - |u| - |q|$ exceeds the sum of the n_i's for all odd i from $2N+1$ to $4N-11$, where n_i is the number of non-red edges of t_i plus the number of κ-edges between t_1 and t_{4N-7} minus 3. By Lemma 11.19 each of these n_i is greater than $||W||/(61k)$. The number of κ-edges between t_1 and t_{4N-7} is greater than 3, so

$$n(\Delta) - n(E) \geq ||W||/(61k).$$

This completes the proof of our lemma. $\square$

The following proposition gives the upper bound for the Dehn function of the group $G_N(\mathcal{S})$.

Proposition 11.1 *If $w = 1$ in $G_N(\mathcal{S})$ then there exists a van Kampen diagram $\Delta(w)$ over $\mathcal{P}_N(\mathcal{S})$ with boundary label w, area $< c_1 \cdot T(c_2 \cdot |w|)^4$ and diameter $< c_1' \cdot T(c_2' \cdot |w|)^3$, for some positive constants c_1, c_2 and c_1', c_2'.*

Proof. By the van Kampen Lemma there exists a reduced diagram Δ over $\mathcal{P}_N(\mathcal{S})$ with boundary label w and minimal possible number of hubs. Then we can apply the snowman decomposition and obtain three sequences of diagrams E_1,...,E_s, Γ_1,...,Γ_{s-1}, Π_1,...,Π_{s-1}, with the following properties:

(S1) $E_s = \Delta$.

(S2) E_1 contains no hubs.

(S3) For every $i = 1, ..., s-1$, we have: either (Γ_i, E_i) is a decomposition of type 1 of E_{i+1} (and Π_i is empty); or (Γ_i, E_i, Π_i) is a decomposition of type 2 of E_{i+1} (in particular, Π_i is a computational disc).

This implies that we can glue together the diagrams $E_1, \Gamma_1, ..., \Gamma_{s-1}, \Pi_1, ..., \Pi_{s-1}$ so as to obtain a diagram with boundary label w. Therefore in order to find the upper bound for the area of Δ we need to estimate the sum of the areas of $E_1, \Gamma_1, ..., \Gamma_{s-1}, \Pi_1, ..., \Pi_{s-1}$.

Property (D5) of decompositions of type 1, and part 2 of Lemma 11.21 show that for every $i = 1, ..., s-1$,

$$n(E_i) < n(E_{i+1}).$$

Therefore

$$n(E_i) \leq |w|, \ i = 1, ..., s,$$

and $s < |w|$.

The diagram E_1 and for every i the diagram Γ_i contain no hubs. So by Lemma 8.1,

$$\text{area}(\Gamma_i) \le C_1 n(\Gamma_i)^3$$

for some constant C_1 and

$$\text{area}(E_1) \le 2C_1 n(E_1)^3.$$

We also know that $n(E_1) \le |w|$ and by Lemma 11.21 and property (D5), $n(\Gamma_i) \le C_2 n(E_{i+1}) \le C_2|w|$, for $i = 1, ..., s-1$ and some constant C_2. Therefore

$$\begin{array}{l} \text{area}(E_1) + \text{area}(\Gamma_1) + \ldots + \text{area}(\Gamma_{s-1}) \ \le \\ C_1 n(E_1)^3 + C_1 n(\Gamma_1)^3 + C_1 n(\Gamma_2)^2 + \ldots \\ +C_1(\Gamma_{s-1})^3 \ \le \\ C_1 C_2 s|w|^3 \le C_3|w|^4. \end{array} \tag{31}$$

for some constant C_3.

It remains to estimate the sum of the areas of discs Π_i. We can assume that each Π_1 corresponds to a minimal area computation of $\mathcal{S}$ connecting some word W_i and W_0. Therefore the area of each Π_i does not exceed "big O" of the area of the corresponding computation. By Proposition 4.1 the area of each Π_i does not exceed $C_4 T(||W_i||)^4 + C_5 n(\Pi_i)^3$ for some constants C_4 and C_5.

$$\begin{array}{l} \text{area}(\Pi_1) + \text{area}(\Pi_2) + \ldots + \text{area}(\Pi_{s-1}) \ \le \\ C_4(T(||W_1||)^4 + \ldots + T(||W_{s-1}||)^4)) + \\ C_5(n(\Pi_1)^3 + n(\Pi_2)^3 + ... + n(\Pi_{s-1})^3) \ \le \\ C_4 T(||W_1|| + \ldots + ||W_{s-1}||)^4 + C_5(n(\Pi_1)^3 + ... + n(\Pi_{s-1})^3). \end{array}$$

Here we used the superadditivity of the function $T(n)^4$. For every i from 1 to $s-1$, if (Γ_i, E_i, Π_i) is a decomposition of type 2 of E_{i+1} then by Lemma 11.21 (2) we have

$$||W_i|| \le C_6(n(E_{i+1}) - n(E_i))$$

for some constant C_6. Therefore

$$||W_1|| + ... + ||W_{s-1}|| \le C_6 n(\Delta) = C_6|w|.$$

By Lemma 11.20 $n(\Pi_i) \le C_8 n(E_{i+1}) \le C_8|w|$ for some constant C_8. Therefore

$$n(\Pi_1)^3 + ... + n(\Pi_{s-1})^3 \le C_9|w| \cdot |w|^3 = C_9|w|^4.$$

Thus

$$\begin{array}{l} \text{area}(\Pi_1) + \text{area}(\Pi_2) + \ldots + \text{area}(\Pi_{s-1}) \ \le \\ C_4 T(C_6|w|)^4 + C_{10}|w|^4. \end{array} \tag{32}$$

for some constant C_{10}.

Combining (31) and (32) we get

$$\text{area}(\Delta) \le C_3|w|^4 + C_4 T(C_6|w|)^4 + C_{10}|w|^4 \le C_{11} T(C_{12}|w|)^4.$$

for some constants C_{11} and C_{12}. This completes the area part of the proposition.

Now let us estimate the diameter $d(\Delta)$. If $\Delta = E_1$ that is if Δ has no hubs then we can apply Lemma 8.1 and conclude that $d(\Delta) \le O(|w|)$. If Δ contains hubs, we can use the snowman decomposition and obtain three sequences of diagrams E_1,..., E_s, Γ_1, ...,Γ_{s-1}, Π_1,...,Π_{s-1} as above. Let $d(\Gamma_i)$, $d(E_i)$, $d(\Pi_i)$ be the diameters of the correspomding diagrams. It is easy to see that

$$d(\Delta) \le \begin{array}{l} \max_{i=1,\dots,s-1}\{d(E_1) + \sum_{j=1}^{s-1} |\partial(\Gamma_j)| + \sum_{j=1}^{s-1} |\partial(\Pi_{s-1})|, \\ d(\Gamma_i) + |\partial(E_1)| + \sum_{j=1}^{s-1} |\partial(\Gamma_j)| + \sum_{j=1}^{s-1} |\partial(\Pi_j)|, \\ |d(\Pi_i)| + |\partial(E_1)| + \sum_{j=1}^{s-1} |\partial(\Gamma_j)| + \sum_{j=1}^{s-1} |\partial(\Pi_j)|\}. \end{array} \tag{33}$$

Indeed, take a vertex inside the diagram Δ' obtained by gluing together E_1, all Γ_i's and all Π_i's. It belongs to one of these blocks. Take the shortest path to the boundary of this block, then we can get to the boundary of Δ' going along the boundaries of the blocks. The resulting path will have length bounded by the right part of (33).

We already know the upper bounds for $|\partial(\Gamma_j)|$ and $|\partial(\Pi_j)|$. By Lemma 11.21 these do not exceed $O(n(\Delta))$. Since $s \le n(\Delta)$, the sums $\sum_{j=1}^{s-1} |\partial(\Gamma_j)|$ and $\sum_{j=1}^{s-1} |\partial(\Pi_j)|$ do not exceed $O(n(\Delta)^2)$. By Lemma 8.1, the diameters of diagrams E_1 and Γ_i ($i = 1, ..., s-1$) do not exceed "big O" of their perimeters. By Lemma 11.21 these perimeters are smaller than $O(n(\Delta))$. Thus the diameters of E_1 and Γ_i do not exceed $O(n(\Delta))$. By Proposition 10.1, the diameter of the disc Π_i does not exceed $O(T(|\partial(\Pi_i)|)^3)$. We also know that $|\partial(\Pi_i)| \le O(n(\Delta))$. Therefore the diameter of Π_i does not exceed $C_1' T(C_2'|w|)^3$ for some constants C_1' and C_2'. Combining this information with (33) we conclude that

$$d(\Delta) \le C_1' T(C_2'|w|)^3 + C_3' n^2$$

for some constant C_3. Since $T(n) \ge O(n)$, we deduce that

$$d(\Delta) \le c_1' T(c_2'|w|)^3$$

for some constants c_1' and c_2' as desired.

The proposition is proved.

□

12 The Lower Bound

The goal of this section is to prove that the group $G_N(\mathcal{S})$ simulates the machine $\mathcal{S}$, that the Dehn function of $G_N(\mathcal{S})$ is bounded below by a function equivalent to $T(n)^4$, and that the smallest isodiametric function is bounded below by a function equivalent to $T(n)^3$.

We start with the following lemma.

Lemma 12.1 *Fix an admissible word W for $\mathcal{S}$. We call a diagram Δ over the presentation $\mathcal{P}_N(\mathcal{S})$* quasi-sector *for W if the boundary of Δ can be represented in the form $b_1 s_1 b_2^{-1} s_2^{-1}$ where b_1 is the top path of a κ_1-band $\mathcal{B}_1$, b_2 is the bottom path of a κ_2-band $\mathcal{B}_2$, and both bands start on s_2; $\mathrm{Lab}(s_1) = \kappa_1 W \kappa_2$, $\mathrm{Lab}(s_2) = \kappa_1 W_0 \kappa_2$; all hubs of Δ are on $\mathcal{B}_1$.*

1. Suppose that Δ is a quasi-sector for W which either has the smallest possible area among all quasisectors for W or has the smallest length of $\mathcal{B}_1$ among all quasi-sectors for W. Then Δ is a sector.

2. Suppose that Δ is a quasi-sector for W. Then the label of $\mathbf{bot}(\mathcal{B}_2)$ is the history word of a computation of $\mathcal{S}$ connecting W and W_0.

Proof. Suppose that $\mathcal{B}_1$ does not have hubs. Then since s_1 and s_2 do not have Θ-edges, every maximal Θ-band which starts on b_1 ends on b_2 and every maximal Θ-band which starts on b_2 ends on b_1. This implies that s_1 is the top path of a Θ-band starting on b_1 and ending on b_2. Thus by definition, Δ is a sector.

Now suppose that $\mathcal{B}_1$ contains hubs. Let us number these hubs $\pi_1, ..., \pi_n$ along $\mathcal{B}_1$. Let π_i be one of these hubs. The path b_1 contains exactly one κ_{N+1}-edge from $\partial(\pi_i)$. Let $\mathcal{K}_i$ be the maximal κ_{N+1}-band of Δ starting on this edge. Since Δ does not have hubs outside $\mathcal{B}_1$, each $\mathcal{K}_i$ ends on the boundary of some hub π_j. Since κ_{N+1}-bands cannot intersect, it is easy to see that there must exist a number i from 1 to $n-1$ such that $\mathcal{K}_i$ ends on π_{i+1}.

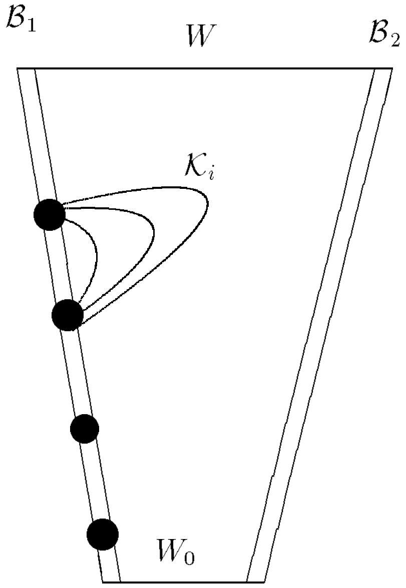

Figure 19.

Consider the subdiagram Σ of Δ bounded by $\mathcal{B}_1$, $\mathcal{K}_i$, and the contours of the two hubs π_i and π_{i+1} such that Σ does not contain π_i and π_{i+1} (see Fig. 19). Let $\partial(\Sigma) = b_1' s_2' (b_2')^{-1} (s_1')^{-1}$ be the corresponding division of the boundary of Σ where $b_1' \subset b_1$, $s_1' \subset \partial(\pi_i)$, $s_2' \subset \partial(\pi_{i+1})^{-1}$, $b_2' \subset \mathbf{bot}(\mathcal{K}_i)$. The diagram Σ does not contain hubs and has $N+1$ κ-bands $\mathcal{T}_1, ..., \mathcal{T}_{N+1}$, connecting edges on s_1 with edges on s_2 where $\mathcal{T}_1 \subset \mathcal{B}_1$, $\mathcal{T}_{N+1} = \mathcal{K}_i$. An argument similar to that used in the proof of Lemma 11.1 shows that Σ is a union of N sectors and inverses of sectors corresponding to the same computation C. By Proposition 9.1 we can assume that these are computational sectors. As in the proof of Lemma 11.1 consider the computational disc Π corresponding to the computation C. We can identify Σ with a part of Π and π_i with the hub of Π. Let $\Gamma = \Pi - \Sigma - \pi_i$. Cut Δ along the path $(b_1')^{-1}\bar{s}_1'b_2'$ where $\bar{s}_1^{-1}s_1 = \partial(\pi_i)$, and glue in the resulting hole the diagrams Γ and Γ^{-1}. Then Π becomes a subdiagram of the resulting diagram Δ'. Then again as in the proof of Lemma 11.1 we notice that the boundary label of Π is the same as the boundary label of the hub, so we can replace the subdiagram Π in Δ' by the hub π_i. After that π_i and π_{i+1} will form a cancellable pair of cells, so we can cancel them. Let $\bar{\Delta}$ be the resulting diagram. Let $\bar{\mathcal{B}}_1$ be the κ_1-band in $\bar{\Delta}$ connecting s_1 and s_2. Finally let $\tilde{\Delta}$ be the subdiagram of $\bar{\Delta}$ bounded by $\bar{\mathcal{B}}_1$, s_1, s_2, $\mathcal{B}_2$. It is easy to see that the area of $\tilde{\Delta}$ is smaller than the area of Δ and the length of $\bar{\mathcal{B}}_1$ is smaller than the length of $\mathcal{B}_1$. Indeed, $\tilde{\Delta}$ is obtained from Δ by removing N sectors of Π and two hubs and inserting N sectors of Π. Therefore $\text{area}(\tilde{\Delta}) \leq \text{area}(\Delta) - 2$; the band $\bar{\mathcal{B}}_1$ is obtained from the band $\mathcal{B}_1$ by removing two hubs, so the $|\bar{\mathcal{B}}_1| = |\mathcal{B}_1| - 2$. This contradicts the assumption that Δ is a quasi-sector for W which either has the smallest possible area among all quasisectors for W or has the smallest length of $\mathcal{B}_1$ among all quasi-sectors for W. This comtradiction proves part 1 of the lemma.

In order to prove part 2 let Δ be a quasi-sector for W. Then reducing the number of hubs of $\mathcal{B}_1$ as above we can transform Δ into a sector $\tilde{\Delta}$ with W as the label of top path. By Proposition 9.1 we can assume that $\tilde{\Delta}$ is a computational sector. Notice that our transformations do not affect $\mathcal{B}_2$. Therefore the label of $\mathbf{bot}(\mathcal{B}_2)$ is the history word of the computation corresponding to the sector $\tilde{\Delta}$. □

The next proposition shows that the group $G_N(\mathcal{S})$ simulates the machine $\mathcal{S}$ and provides a lower bound for the Dehn function of this group.

Proposition 12.1 *1. For every admissible word W of the machine $\mathcal{S}$ there exists a computation of $\mathcal{S}$ connecting W and W_0 if and only if $K(W) = 1$ in the group $G_N(\mathcal{S})$.*

2. The Dehn function of $G_N(\mathcal{S})$ is bounded below by a function equivalent to $T(n)^4$.

Proof. If there exists a computation of $\mathcal{S}$ connecting W and W_0 then the computational disc corresponding to this computation has boundary label $K(W)$, so $K(W) = 1$ in the group $G_N(\mathcal{S})$.

Conversely suppose that $K(W) = 1$ in $G_N(\mathcal{S})$. Then there exists a van Kampen diagram Δ over the presentation $\mathcal{P}_N(\mathcal{S})$ with boundary label $K(W)$. Our goal is to prove that there exists a computation connecting W and W_0 and to estimate the area and diameter of Δ from below. Thus we assume that the area of Δ is minimal. Then Δ is reduced. Let e_i, $i = 1, ..., 4N$ be $\kappa_i^{\pm 1}$-edges on $\partial(\Delta)$, we assume that e_1 precedes e_2,..., precedes e_{4N} on $\partial(\Delta)$. Let $\mathcal{K}_i$, $i = 1, ..., 2N$ be the maximal κ_i-band starting on e_i. Since the contour of Δ contains two κ_1 edges and two κ_2-edges, $\mathcal{K}_1$ and $\mathcal{K}_2$ intersect. Therefore Δ contains hubs.

Remark. Now it is tempting to use the graph theoretic lemmas from the previous section, to deduce that Δ contains just one hub, and then prove that Δ is a disc. But unfortunately we cannot use the lemmas from the previous section because we cannot assume that Δ has minimal number of hubs: this assumption may contradict the minimal area assumption or the minimal diameter assumption. Thus our proof proceeds in a different direction.

Notice that for every i from 1 to $2N$, Δ contains only two mutually inverse κ_i-bands. Therefore each hub in Δ belongs to all $\mathcal{K}_i$.

Let $\pi_1,...,\pi_m$ be all the hubs in Δ, numbered along $\mathcal{K}_1$. Let π_n be the first of these hubs appearing in $\mathcal{K}_2$. Consider the subdiagram Γ of Δ bounded by $\mathcal{K}_1$, $\partial(\Delta)$, $\mathcal{K}_2$ and $\partial(\pi_n)$ such that $\pi_n \notin \Gamma$. It is easy to see that Γ is a quasi-sector for W.

By Lemma 12.1 there exists a computational sector Σ with top path labelled by $\kappa_1 W \kappa_2$ and the bottom path labelled by $\kappa_1 W_0 \kappa_2$ such that area$(\Sigma) \leq$ area(Γ). By Proposition 9.1 there exists a computation of $\mathcal{S}$ connecting W and W_0. This proves statement 1 of our proposition.

Let $S(n)$ be the space function of the Turing machine M. We know that $S(n)$ is equivalent to $T(n)$. Using the notation from Proposition 4.1 let us assume that $W = \sigma(c)$ where c is an accepted configuration of M of length $\leq n$ such that any smallest space accepting computation of M for the configuration c has space $S(n)$. Then by Proposition 4.1, any computation of $\mathcal{S}$ connecting W and W_0 has area exceeding $CS^4(n)$ and time exceeding $C'S^3(n)$ for some positive constants C, C'. Therefore by Proposition 9.1 the area of Σ exceeds $O(S^4(n))$ and the diameter exceeds $O(S^3(n))$.

Since area$(\Delta) \geq$ area$(\Gamma) \geq$ area(Σ), the area of Δ exceeds $O(S(n)^4)$. Therefore $S(n)^4$ is equivalent to a lower bound for the Dehn function of $G_N(\mathcal{S})$. Since $S^4(n)$ is equivalent to $T(n)^4$, $T(n)^4$ is also equivalent to a lower bound of the Dehn function of $G_N(\mathcal{S})$. $\square$

The next proposition provides a lower bound for isodiametric functions of $G_N(\mathcal{S})$.

Proposition 12.2 *Every isodiametric function of $G_N(\mathcal{S})$ is bounded below by a function equivalent to $T(n)^3$.*

Proof. Let $\mathcal{T}(\backslash)$ be the "time function" of the machine $\mathcal{S}$, that is $\mathcal{T}(n)$ is the smallest number such that for every admissible word W of length $\leq n$, for which there exists a computation connecting it with W_0, the smallest length of this computation is $\leq \mathcal{T}(n)$. By the choice of $\mathcal{S}$, $\mathcal{T}(n)$ is equivalent to $T(n)^3$.

Let us fix a number $n \geq 1$ and let W be an adsmissible word for the S-machine $\mathcal{S}$, $|W| \leq n$. Suppose that there exists a computation connecting W and W_0 and the smallest length of this computation is $t(n)$. Then there exist a diagram (for example, computational disc) over $\mathcal{P}_N(\mathcal{S})$ with boundary label $K(W)$. Let Δ be any (not necessarily reduced) diagram over the presentation $\mathcal{P}_N(\mathcal{S})$ with boundary label $K(W)$. Assume that Δ has the smallest diameter among all such diagrams.

By Lemma 7.1 the operation of removing κ-annuli does not increase the diameter of Δ. So we can assume that Δ does not contain κ-annuli. This implies as before that every hub in Δ belongs to every κ-band and that Δ contains 2 mutually inverse maximal κ_i-bands for every $i = 1, ..., 2N$. As in the proof of Proposition 12.2, let e_i, $i = 1, ..., 4N$ be $\kappa_i^{\pm 1}$-edges on $\partial(\Delta)$, we assume that e_1 precedes e_2,..., precedes e_{4N} on $\partial(\Delta)$. Let $\mathcal{K}_i$, $i = 1, ..., 2N$ be the maximal κ_i-band starting on e_i. Since the contour of Δ contains two κ_1 edges and two κ_2-edges, $\mathcal{K}_1$ and $\mathcal{K}_2$ intersect. Therefore Δ contains hubs.

Let $\pi_1,...,\pi_m$ be all the hubs in Δ, numbered along $\mathcal{K}_1$. Let π_n be the first of these hubs appearing in $\mathcal{K}_2$. As before, consider the subdiagram Γ of Δ bounded by $\mathcal{K}_1$, $\partial(\Delta)$, $\mathcal{K}_2$ and $\partial(\pi_n)$ such that $\pi_n \notin \Gamma$.

Let $\mathcal{B}_1$ (resp. $\mathcal{B}_2$) be the κ_1-band (resp. κ_2-band) of Γ starting on the contour of Δ. Let f_i be the end edge of $\mathcal{B}_i$ ($i = 1, 2$). Let v be the initial vertex of the κ_2-edge f_2 and let p be the shortest path connecting v with the boundary of Δ.

In every pair of mutually inverse maximal Θ-bands in Δ, call one band *positive* and the other one *negative*. Consider the set Ω of all positive maximal Θ-bands of Δ intersecting $\mathcal{B}_2$. At most $|\partial(\Delta)|$ of these bands start and end on the boundary of Δ. Therefore $|\Omega| - |\partial(\Delta)|$ bands in Ω are Θ-annuli. Our goal is to calculate the number of Θ-annuli in Ω which contain v in their inside diagrams. It is clear that the length of p cannot be smaller than this number.

Notice that given Δ, Γ, $\mathcal{B}_2$, v and $|p|$ are determined uniquely. Without loss of generality we assume that Δ has the smallest possible pair $(|p|, |\mathcal{B}_2|)$ (in the lexicographic order) among all diagrams with boundary label $K(W)$.

Suppose that p contains at least two $\kappa_2^{\pm 1}$-edges of $\mathcal{B}_2$. We shall show that this leads to a contradiction.

Then p contains two edges e_1 and e_2 which are common edges of $\mathcal{B}_2$ or their inverses and such that between e_1 and e_2, p does not intersect the median of $\mathcal{B}_2$. Let t_1 be the initial vertex of e_1 and let t_2 be one of the vertices of e_2 which belong to the same side of $\mathcal{B}_2$ (top ot bottom) as t_1. Then there exists a reduced path q which connects t_1 and t_2 and is a subpath of $\mathbf{top}(\mathcal{B}_2)^{\pm 1}$ or $\mathbf{bot}(\mathcal{B}_2)^{\pm 1}$.

Claim. Lab(q) is a reduced word.

Indeed, suppose that q contains a subpath ee' where e and e' are edges with mutually inverse $\Theta^{\pm 1}$-labels. Then the corresponding cells ρ_1 and ρ_2 of $\mathcal{B}_2$ form a reducible pair. The common edge of this pair does not belong to $p^{\pm 1}$ because p does not have κ_2-edges from $\mathcal{B}_2$ between e_1 and e_2. Therefore reducing this pair does not affect the length of p (if this edge belonged to p then reducing this pair of cells would make the shortest path connecting v and $\partial(\Delta)$ longer). When we reduce ρ_1 and ρ_2, we get a diagram with smaller pair $(|p|, |\mathcal{B}_2|)$ which contradicts our choice of Δ.

Now let p' be the portion of p which connects t_1 and t_2. Then $(q^{-1}p')^{\pm 1}$ bounds a subdiagram of Δ. Therefore Lab(p') = Lab(q) in $G_N(\mathcal{S})$. There exists a homomorphism from $G_N(\mathcal{S})$ to the free group generated by Θ which takes all non-Θ-letters to 1. Therefore if we remove all non-Θ-letters from p', we get a word U which is freely equal to Lab(q). Since by Claim Lab(q) is reduced, $|U| \geq |q|$. Notice that Lab(p') contains a non-Θ-edge e_1. Therefore $|p'|$ is strictly greater than $|q|$. Therefore we can substitute p' by q in p and get a shorter path connecting v with $\partial(\Delta)$, a contradiction with the choice of p. This contradiction shows that p contain at most one $\kappa_2^{\pm 1}$-edge of $\mathcal{B}_2$.

Suppose that p contains one $\kappa_2^{\pm 1}$-edge e of $\mathcal{B}_2$. One of the vertices v' of e belongs to the same side (top or bottom) of $\mathcal{B}_2$ as v. Let q be the subpath of $\mathbf{bot}(\mathcal{B}_2)^{-1}$ connecting v and v'. As in the proof of Claim one can show that Lab(q) is a reduced word and that the part p' of the path p connecting v and v' is not shorter than q. Let $\bar{v}$ be the end vertex of the κ_2 edge f_2 and let $\bar{v}'$ be the other vertex of e. Then the vertices $\bar{v}$ and $\bar{v}$ are connected by a subpath $\bar{q}$ of $\mathbf{top}(\mathcal{B}_2)^{-1}$. Notice that as before we can assume that the word Lab$(\bar{q})$ is reduced. So the lengths of q and $\bar{q}$ are equal. Thus we can substitute the subpath p' of p by $f_2\bar{q}$ and get another path, say, $\bar{p}$, which connects v with the boundary of Δ, has length not bigger than the length of p, and contains only one common κ_2-edge with $\mathcal{B}_2$, the edge f_2. Thus we can assume without loss of generality that the path p has these properties.

Now if $\mathcal{B}_2$ contains reducible pairs of cells we can cancel them without affecting the length of p.

In the case when p does not have common $\kappa^{\pm 1}$-edges with $\mathcal{B}_2$, we can reduce $\mathcal{B}_2$ without changing the length of p.

Thus we can assume that $\mathcal{B}_2$ is reduced. Now we can apply Lemma 7.5 and deduce that no Θ-band from Ω can have two intersections with $\mathcal{B}_2$ (otherwise $\mathcal{B}_2$ would not be reduced). Thus the number of Θ-bands in Ω is $|\mathcal{B}_2|$.

Every Θ-annulus in Ω must contain v in its inner subdiagram. Indeed, the bottom path of $\mathcal{B}_2$ crosses this annulus exactly once, so the index of the median of this annulus (which is by definition a simple closed curve) with respect to the point v is 1.

Therefore the number of annuli in Ω is at least $|\mathcal{B}_2|-|\partial(\Delta)|$. Let us reduce the diagram Δ by first reducing the band $\mathcal{B}_1$ and then reducing other reducible pairs of cells. Let $\bar{\Gamma}$ be the reduced diagram obtained as a result of this process. Then $\bar{\Gamma}$ is a quasi-sector. By Lemma 8.4 the band $\mathcal{B}_2$ does not change during the reduction process in Γ. Therefore by Lemma 12.1 Lab$(\mathcal{B}_2)$ is a history of a computation connecting W with W_0. Therefore by $|\mathcal{B}_2| \geq \mathcal{T}(n)$. Therefore the number of Θ-annuli in Δ containing v is not smaller than $\mathcal{T}(n)-|K(W)| \geq \mathcal{T}(n)-4Nn-4N$. Hence the length of p is at least $\mathcal{T}(n)-4Nn-4N$. Since the length of $\partial(\Delta)$ does not exceed $4Nn+4N$, we deduce that the isodiametric function of $G_N(\mathcal{S})$ is $\succ \mathcal{T}(n) \equiv T(n)^3$. $\square$

13 Proof of Theorem 1.3

The proof of Theorem 1.3 can be completed as follows. Let $L \subseteq X^+$ be a language recognized by a Turing machine with time function $T(n)$. Then there exists a symmetric nondeterministic Turing machine M, with all the properties of Lemma 3.1, which recognizes the language L. Take any $N \geq 6$ and construct the group $G_N(\mathcal{S})$. Let $D(n)$ be the Dehn function of this group. Then by Proposition 11.1, $D(n) \leq C_1 T(C_2 n)^4$, for some constants C_1 and C_2. By Proposition 12.1, $D(n) > c_1 T(c_2 n)^4$, for some positive constants c_1, c_2. Therefore $D(n)$ is equivalent to $T(n)^4$.

Similarly we can prove that the smallest isodiametric function of Δ is equivalent to $T(n)^3$.

Let w be an input word for the machine M and c_w be the corresponding input configuration of the machine M. Then by Proposition 12.1 the correspondence $w \to K(\sigma(c_w))$ satisfies the conditions of Theorem 1.3, in particular $w \in L$ if and only if $K(\sigma(c_w)) = 1$ in $G_N(\mathcal{S})$. Theorem 1.3 is proved.

Theorem 1.2 follows from Theorem 1.3 and Theorem 1.1.

14 Proof of Corollary 1.2

This corollary almost immediately follows from Theorem 1.3 and its proof. Take any Turing machine M which accepts a non-recursive language. We can assume that M satisfies the conditions of Lemma 3.1. Using Proposition 4.1 we construct the S-machine $\mathcal{S} = \mathcal{S}(M)$ for which there is no algorithm deciding whether there exists a computation of $\mathcal{S}$ connecting a given admissible word W with the fixed admissible word W_0. Now take $N \geq 6$ and consider the group $G'_N(\mathcal{S})$ given by the presentation $\mathcal{P}'_N(M)$ which is obtained from $\mathcal{P}_N(M)$ by removing the hub relations.

In the group $G'_N(\mathcal{S})$ a word $K(W)$ where W is an admissible word for $\mathcal{S}$ is conjugate with $K(W_0)$ if and only if W is connected with W_0 by a computation of $\mathcal{S}$. Indeed, if such a computation exists then by Proposition 10.1 there exists a disc with boundary label $K(W)$. By removing the hub from this disc we obtain an annular diagram over the presentation $\mathcal{P}'_9(M)$ which shows that $K(W)$ is conjugate to $K(W_0)$ (see [20]). On the other hand if $K(W)$ is conjugate to $K(W_0)$ then there exists an annular diagram over $\mathcal{P}'_N(\mathcal{S})$ with boundary labels $K(W)$ (the outer contour) and $K(W_0)$ (the inner contour). We can glue the hub in the hole of this annular diagram and obtain a diagram with one hub over $\mathcal{P}_N(\mathcal{S})$ which by definition is a disc. By Proposition 10.1 then W is connected with W_0 by a computation of $\mathcal{S}$.

Since the language accepted by M is not recursive, $G'_9(\mathcal{S})$ has undecidable conjugacy problem. It remains to notice that by Lemma 8.1 the Dehn function of $G'_9(M)$ is at most $O(n^3)$ (indeed, the van Kampen diagrams over $\mathcal{P}'_N(\mathcal{S})$ are precisely the van Kampen diagrams over $\mathcal{P}_N(\mathcal{S})$ without hubs).

Notice also that the group we constructed has linear isodiametric function by Lemma 8.1. $\square$

References

[1] S. Aanderaa. A proof of Higman's embedding theorem using Britton extensions of groups. In *Word Problems, Decision Problems and the Burnside problem in group theory*, pages 1–18, North-Holland, Publ. Corp., Amsterdam, London, 1973.

[2] J. Alonso, Inégalités isopérimétriques et quasi-isométries, C. R. Acad. Sci. Paris Série 1, 311 (1990), 761-764.

[3] G. Baumslag, C. F. Miller III, and H. Short. Isoperimetric inequalities and the homology of groups. Invent. Math., 113:531–560, 1993.

[4] J.C. Birget, "Time-complexity of the word problem for semigroups, and the Higman Embedding Theorem", to appear in IJAC.

[5] R. P. Brent, "Fast multiple-precision evaluation of elementary functions", J. of the ACM 23 (1976), 242-251.

[6] M. Bridson, "Fractional isoperimetric inequalities.", preprint, 1995

[7] D E Cohen, "Isodiametric and isoperimetric inequalities for group representations" IJAC 1 (1992) 315-320.

[8] W.J. Floyd, A.H.M. Hoare, R.C. Lyndon, "The word problem for geometrically finite groups", Geomtriae Dedicata 20 (1986), 201-207.

[9] S. M. Gersten, "Dehn functions and l_1-norms for finite presentations", in *Algorithms and Classification in Combinatorial Group Theory* (G. Baumslag, C. F. Miller, editors), MSRI Publications 23 (1992), Springer-Verlag.

[10] S. M. Gersten, "Isoperimetric functions of groups and exotic cohomology", in *Combinatorial and Geometric Group Theory,* (A.J. Duncan, N.D.Gilbert, J. Howie, editors), Edinburgh, 1993, London Mathematical Soc. Lecture Notes Series, 204, 1995.

[11] S M Gersten, "The double exponential theorem for isodiametric and isoperimetric functions", IJAC 1 (1991) 321-328.

[12] M. Gromov, "Hyperbolic groups", in: *Essays in group theory* (S.M. Gersten, editor), MSRI Series 8, Springer-Verlag, 1987.

[13] M. Gromov, "Asymptotic invariants of infinite groups", in: *Geometric Group Theory, Volume 2* (G. A. Niblo and M. A. Roller, eds.), London Mathematical Society Lecture Notes Series 182, 1993.

[14] V. Guba and M. Sapir, *Diagram groups*, to appear in Memoirs of AMS, 1997.

[15] V. Guba and M. Sapir, *The Dehn function of a free product of non-trivial groups is superadditive*, preprint, 1996.

[16] J. Hopcroft, J. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley (1979).

[17] O. Kharlampovich, M. Sapir, *Algorithmic problems in varieties*, IJAC, 1995.

[18] J. van Leeuwen (ed.), *Handbook of Theoretical Computer Science*, Vol. A, Elsevier and MIT Press (1990).

[19] H.R. Lewis, Ch. Papadimitriou, "Symmetric space-bounded computation", *Theoretical Computer Science* 19 (1982) 161-187.

[20] R. C. Lyndon and P. E. Schupp. *Combinatorial group theory.* Springer-Verlag, 1977.

[21] K. Madlener, F. Otto, "Pseudo-natural algorithms for the word problem for finitely presented monoids and groups", *J. Symbolic Computation* 1 (1989), 383-418.

[22] Charles F. Miller III and Paul E. Schupp, "The geometry of Higman-Neumann-Neumann extensions", Comm. Pure and Appl. Math., XXVI (1973), 787-802.

[23] A. Yu. Ol'shanskii. *The geometry of defining relations in groups*, Nauka, Moscow, 1989.

[24] A. Yu. Ol'shanskii. *On distortion of subgroups in finitely presented groups.* preprint, 1997.

[25] Ch. Papadimitriou, *Computational Complexity*, Addison-Wesley (1994).

[26] J. Rotman. *An Introduction to the Theory of Groups.* Allyn & Bacon, third edition, 1984.

Mark V. Sapir
Department of Mathematics and Statistics
University of Nebraska-Lincoln
http://www.math.unl.edu/~msapir

Jean-Camille Birget
Department of Computer Science
University of Nebraska-Lincoln
birget@cse.unl.edu

Eliyahu Rips
Department of Mathematics
Hebrew University of Jerusalem
rips@math.huji.ac.il