ADVANCES IN MATHEMATICS 46, 305–329 (1982)

The Complexity of the Word Problems for Commutative Semigroups and Polynomial Ideals*

ERNST W. MAYR AND ALBERT R. MEYER

Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

Any decision procedure for the word problems for commutative semigroups and polynomial deals inherently requires computational storage space growing exponentially with the size of the problem instance to which the procedure is applied. This bound is achieved by a simple procedure for the semigroup problem.

1. INTRODUCTION

The word problem for commutative semigroups is effectively decidable. In fact for any fixed finitely presented commutative semigroup, testing equivalence of two words over the generators reduces to evaluating a linear form and is computationally trivial, i.e., solvable in real-time on a Turing machine [11, 23, 42]. The *uniform* word problem, in which the defining equations as well as the words are regarded as an instance of the problem, is also effectively decidable for commutative semigroups. This was first explicitly noted by [8, 25], though in retrospect this result can be seen to be a special case of results of [14–16, 22] on testing membership in polynomial ideals. The known procedures for deciding the uniform word problem, however, require considerably more effort to carry out. We show in this paper that this is inevitable: *any* decision procedure requires an amount of storage space for intermediate results of computation which grows exponentially with the size of the problem instance to which the procedure might be applied. We also show that this exponential bound on the complexity of decision procedures is achievable by a naive search for a derivation of one word from the other.

Results establishing the inherent computational complexity of decidable problems are the natural quantitative refinement of classical results in algebra, logic, and other branches of mathematics distinguishing decidable from undecidable problems. Problems, such as the uniform word problem for

* This work was supported in part by the Deutsche Forschungsgemeinschaft, Grant I3 Ma 870/1-1, by The National Science Foundation, Grant MCS 8010707, and by a grant to the M.I.T. Laboratory for Computer Science by the IBM Corporation.

0001-8708/82/120305-25$05.00/0

commutative semigroups, which are decidable *in principle* but whose complexity is exponential or greater present the same intractability as undecidable problems. To illustrate this, note that undecidability of a problem means that every procedure (e.g., Turing machine) which gives only correct decisions on instances of the undecidable problem must, for infinitely many instances, fail to produce a decision. Exponential complexity of a problem means that every procedure which gives only correct decisions on instances of the complex problem must, for infinitely many instances, take a prohibitive amount of computational resource to produce a decision. In both cases an observer is left waiting on tenterhooks for an answer which will never come in his lifetime.

Computational complexity theory has become a reasonably developed mathematical subject in the past decade. Its basic concept of growth rate of computational resource usage as a function of the size of the input to a procedure is recognized to have much the same robustness at the Church–Gödel–Turing notion of effective procedure [1, 7, 18, 20, 32]. In particular, the property that a problem requires exponential space to decide effectively is invariant over the exact formulation of models of effective procedures or the details of the measure of space required by a procedure. For definiteness, we take Turing machines as a standard model of computation and define the *space* required by a Turing machine on a given input to be the number of work tape squares visited by the head of the machine during the computation on that input.

Clearly it requires more computational effort to deal with larger problem instances, so complexity is usually measured *relative* to the size of a problem instance. Let S be some finite set of symbols each taken to be of unit size, $\mathscr{P}$ some finite commutative semigroup presentation, and α, β two words over S. The uniform word problem for commutative semigroups, abbreviated CSG, is

$$\{(\alpha, \beta, \mathscr{P});\ \text{equivalence of } \alpha \text{ and } \beta \text{ is derivable from } \mathscr{P}\}$$

(more detailed definitions appear in Section 2). Any triple $(\alpha, \beta, \mathscr{P})$ is a *CSG problem instance*, and the *size* of $(\alpha, \beta, \mathscr{P})$ is taken to be the length of a list consisting of α, β and the left- and righthand sides of the equations in $\mathscr{P}$, separated by unit size delimiters. It is natural to allow exponential notation in representing words over S. For example, a word consisting of 1003 s's has size 5 because it has a representation in exponential notation of five symbols, namely, s^{1003}. We emphasize, however, that our results are not dependent on this representation; even if we forbade exponential notation and defined the size of problem instances to be their total length our main theorem still holds.

MAIN THEOREM. (a) *There is a constant $c > 0$ and an algorithm*

(Turing machine) which decides CSG and requires space at most 2^{cn} *on any instance of CSG of size n.*

(b) *There is a constant* $\varepsilon > 0$ *such that any algorithm which decides CSG requires space exceeding* $2^{\varepsilon n}$ *on an instance of CSG of size n for infinitely many n.*

This main Theorem appears as Theorems 1 and 2 in Sections 4 and 7, respectively.

CSG is closely related to a basic decision problem of classical algebra, the polynomial ideal word problem *PI*. Let X be some finite set of indeterminates and $p_0,..., p_n \in \mathbb{Q}[X]$.[1] Then *PI* is defined to be

$$\{\langle p_0,..., p_n\rangle;\ p_0 \text{ is in the ideal of } \mathbb{Q}[X] \text{ generated by } p_1,..., p_n\}.$$

We show in Section 3 below that *CSG* is straightforwardly reducible to *PI*. This allows us to appeal to results of [15] on solutions of linear equations over $\mathbb{Q}[X]$ to obtain the *upper* bound on the complexity of *CSG* stated in part (a) of the Main Theorem. (We include a concise version of Hermann's result in the Appendix.) Conversely, the *lower* bound on the complexity of *CSG* given in part (b) of the Main Theorem implies a corresponding lower bound on *PI*:

MAIN COROLLARY. *There is a constant* $\varepsilon > 0$ *such that any algorithm which decides PI requires space exceeding* $2^{\varepsilon n}$ *on an instance of PI of size n for infinitely many n.*

Here an instance of *PI* is the $n + 1$-tuple $\langle p_0,..., p_n\rangle$, and its size is defined to be the sum of the lengths of the coefficients and exponents, written as (quotients of) Arabic numerals, of each of the terms of the polynomials.

The key technical fact on which the proof of the lower bound on space requirements rests is the possibility of faithfully embedding commutative semigroups with "large" finite presentations into commutative semigroups with "small" presentations. In particular, we show how a commutative semigroup with a defining equation of the form

$$s_1 \equiv s_2^{2^{2^n}},$$

which by our conventions has size proportional to 2^n, can be embedded in a commutative semigroup whose presentation (even without use of exponential notation) is of size $O(n)$.

The existence of embeddings into *succinct* presentations is the complexity theoretic analogue of the classical result that every recursively enumerable

[1] $\mathbb{N}$ denotes the set $\{0, 1,...\}$ of nonnegative integers. $\mathbb{Z}$ the set of integers, $\mathbb{Q}$ the set of rationals, and for $n \in \mathbb{N}$, I_n the set $\{1,..., n\}$.

(r.e.) presentation of an *arbitrary*—not necessarily commutative—semigroup is embeddable in a *finitely* presented semigroup. The undecidability of the word problem for arbitrary semigroups follows immediately from this embedding and the existence of sets such as the Halting Problem for Turing machines which are r.e. but undecidable [27, 34]. Thus, the proof of the exponential space lower bounds is similar to a standard undecidability proof by reduction of the Halting Problem.

This pattern of argument is standard in complexity theory, but may be worth reviewing for readers unfamiliar with complexity theory. In Section 5, Lemma 4, we define a complexity theoretic analogue, *ESC*, of the Halting Problem. *ESC* is exponential space complete: it is decidable within exponential space and has the property that all problems decidable by procedures using at most exponential space are *efficiently* reducible to it, in a precise sense defined in Section 2. Elementary diagonal arguments of complexity theory have previously established the existence of sets which are decidable in exponential space but not less space [3, 19]. It follows that *ESC* requires exponential space since it must be as complex as any set which reduces to it. In Section 5, we show that *ESC* is itself efficiently reducible to a version of *CSG* in which presentations contain defining equations of the double exponential form noted above. The key fact about succinct embedding which allows elimination of these large equations is established in Section 6, and we conclude in Section 7 that *ESC* is efficiently reducible to *CSG*, so that *CSG* is itself exponential space complete.

The results described here were presented in preliminary form in [6].

2. Exponential Space, Semi-Thue Systems, and Semigroup Presentations

We first briefly review the few necessary technical definitions from complexity theory. For more complete treatments see [9, 18, 32].

For any finite alphabet S of symbols, let S^* be the set of all finite words over S. A function f: $S_1^* \to S_2^*$ *reduces* a set $A \subseteq S_1^*$ to a set $B \subseteq S_2^*$ providing that

$$\alpha \in A \Leftrightarrow f(\alpha) \in B$$

for all $\alpha \in S_1^*$. If f is computable by a Turing machine which visits at most $\log_2 n$ work tape squares during its computation on any word $\alpha \in S_1^*$ of length $n > 1$, then A is said to be *log-space reducible* to B. (We assume the Turing machine has a read-only input tape and a write-only output tape separate from its work tape.) If in addition the length of $f(\alpha)$ is $O(\text{length}(\alpha))$, then A is *log-lin reducible* to B [31, 40, 41].

The set $B \subseteq S_2^*$ is said to be *decidable in space* g: $\mathbb{N} \to \mathbb{N}$ if there is a Turing machine which accepts B and visits at most $g(n)$ work tape squares during its computations on any word $\beta \in S_2^*$ of length n. B is decidable in *exponential space* if it is decidable in space g, where $g(n) \leqslant c^n$ for some $c > 1$. B is *exponential space complete with respect to log-lin reducibility* if (1) it is decidable in exponential space, and (2) every set which is decidable in exponential space is log-lin reducible to B. If B satisfies condition (2) only, it is said to be *exponential space hard.*

Suppose A is log-lin reducible to B. Then any procedure for deciding B immediately yields a procedure for deciding A which uses essentially the same space. In particular, these is a $k > 0$ such that, given any Turing machine which decides B in space c^n, one can exhibit a Turing machine which decides A in space c^{kn}.

Now suppose B is exponential space hard. An elementary diagonal argument may be used to establish the existence of a set A which is decidable in space say 3^n but not 2^n [3, 19]. Since A is decidable in exponential space, it is log-lin reducible to B. Since A is not decidable in space 2^n, the set B cannot be decidable in space $2^{\varepsilon n}$, where $\varepsilon = 1/k$. Thus to prove an exponential space lower bound on decision procedures for an arbitrary set B it is sufficient to prove that B is exponential space hard with respect to log-lin reducibility. Log-lin reducibility can be shown to be transitive, so to prove that a set B is exponential space hard, it is sufficient to prove that some set A, already known to be exponential space hard, is log-lin reducible to B.

This completes our review of complexity theory; we now give the basic definitions concerning word problems.

Let $S = \{s_1, \ldots, s_v\}$ be a finite alphabet. A *semi-Thue system* over S is given by a finite set $\mathscr{P}$ of productions $l_i \to r_i$, where $l_i, r_i \in S^*$. A word $\beta \in S^*$ *is derived in one step* from $\alpha \in S^*$ (written $\alpha \to \beta(\mathscr{P})$) by application of the production $(l_i \to r_i) \in \mathscr{P}$ iff for some $\gamma, \delta \in S^*$, we have $\alpha = \gamma l_i \delta$ and $\beta = \gamma r_i \delta$. The word α *derives* β iff $\alpha \overset{*}{\to} \beta(\mathscr{P})$, where $\overset{*}{\to}$ is the reflexive transitive closure of $\to$. A sequence $(\alpha_0, \ldots, \alpha_n)$ of words $\alpha_i \in S^*$ with $\alpha_i \to \alpha_{i+1}$ for $i = 0, \ldots, n-1$ is called a *derivation* (of length n) of α_n from α_0 in $\mathscr{P}$.

A *semigroup presentation* or *Thue system* is a symmetric semi-Thue system $\mathscr{P}$, i.e.,

$$(l \to r) \in \mathscr{P} \Rightarrow (r \to l) \in \mathscr{P}.$$

Derivability in a semigroup establishes an equivalence relation $\equiv$ by the rule

$$\alpha \equiv \beta(\mathscr{P}) \underset{\text{def}}{\Leftrightarrow} \alpha \overset{*}{\to} \beta(\mathscr{P}).$$

For semigroups, we also use the notation $l \equiv r(\mathscr{P})$ to denote the pair of productions $(l \to r)$ and $(r \to l)$ in $\mathscr{P}$.

A semi-Thue system $\mathscr{P}$ is *commutative* if

$$(\forall s, s' \in S)[(ss' \to s's) \in \mathscr{P}].$$

If it is understood that $\mathscr{P}$ is a commutative semi-Thue system these commutativity productions are not explicitly mentioned in $\mathscr{P}$ nor is their application within a derivation in $\mathscr{P}$ counted as a step. We remark that commutative semi-Thue systems appear in the literature in two additional equivalent formulations: *vector replacement systems* (VRSs) [21] and *Petri nets* [13, 17, 26, 33]. Finitely presented commutative semigroups are equivalent to *reversible* VRSs or Petri nets [12].

Let $\Phi: S^* \to \mathbb{N}^v$ be the Parikh mapping, i.e., $(\Phi(\alpha))_i$ (also written $\Phi(\alpha, s_i)$) indicates, for every $\alpha \in S^*$ and $i \in I_v$, the number of occurrences of $s_i \in S$ in α.

For a word in a commutative semigroup generated by S the order of the symbols is immaterial, and we shall in the sequel use an exponent notation. For instance, we may denote *abaacba* by a^4b^2c, interchangeably with, say, a^3cb^2a. Let $\alpha, \beta \in C^*$ and $\mathscr{P}$ be a finite set of productions over S. We define size$(\alpha, \beta, \mathscr{P})$ to be the length of the list consisting of representations in exponent notation as above of α, β, and l_i, r_i, for $(l_i \equiv r_i) \in \mathscr{P}$ (omitting pure commutative relations). Exponents are written with Arabic numerals, properly interspersed with delimiters to separate vectors and their components, and each $s_i \in S$ is taken to have unit length.

3. Degree Bounds for Polynomial Ideals

Let X denote the finite set $\{x_1, ..., x_v\}$, and $\mathbb{Q}[X]$ (resp., $\mathbb{Z}[X]$) the (commutative) ring of polynomials with indeterminates $x_1, ..., x_v$ and rational (resp., integer) coefficients. For $p_1, ..., p_w \in \mathbb{Q}[X]$, let $(p_1, ..., p_w) \subseteq \mathbb{Q}[X]$ denote the ideal generated by $\{p_1, ..., p_w\}$, that is,

$$(p_1, ..., p_w) \stackrel{\text{def}}{=} \left\{\sum_{i=1}^{w} g_i p_i;\ g_i \in \mathbb{Q}[X] \text{ for } i \in I_w\right\}.$$

Now let $\mathscr{P} = \{\alpha_i \equiv \beta_i;\ i \in I_w\}$ be any (finite) commutative semigroup presentation with $\alpha_i, \beta_i \in X^*$ for $i \in I_w$. We identify any $\alpha \in X^*$ with the unary monomial $\alpha = x_1^{\Phi(\alpha, x_1)} \cdots x_v^{\Phi(\alpha, x_v)}$, and let $\mathrm{I}_{\mathbb{Q}}(\mathscr{P})$ (resp., $\mathrm{I}_{\mathbb{Z}}(\mathscr{P})$) be the $\mathbb{Q}[X]$-ideal (resp., $\mathbb{Z}[X]$-ideal) generated by $\{\beta_1 - \alpha_1, ..., \beta_w - \alpha_w\}$, i.e.,

$$\mathrm{I}_R(\mathscr{P}) \stackrel{\text{def}}{=} \left\{\sum_{i=1}^{w} g_i(\beta_i - \alpha_i);\ g_i \in R[X] \text{ for } i \in I_w\right\} \qquad \text{for} \quad R = \mathbb{Q}, \mathbb{Z}.$$

The next few lemmas show the connection between *CSG* and the membership problem for ideals in $\mathbb{Z}[X]$ and $\mathbb{Q}[X]$. Also see [39] for part of these results.

LEMMA 1. *If* $\alpha \equiv \beta(\mathscr{P})$, *then* $\beta - \alpha \in \mathrm{I}_{\mathbb{Z}}(\mathscr{P})$.

Proof. Suppose $\alpha = \gamma_0 \to \gamma_1 \to \cdots \to \gamma_n = \beta(\mathscr{P})$ and assume without loss of generality that $n \geqslant 1$. Then, for $m \in I_n$, there are $\delta_m \in X^*$ and $i_m \in I_w$ such that

$$\gamma_{m-1} = \alpha_{i_m}\delta_m \qquad \text{and} \qquad \gamma_m = \beta_{i_m}\delta_m,$$

and hence,

$$\beta - \alpha = \sum_{m=1}^{n} (\beta_{i_m} - \alpha_{i_m})\,\delta_m \in \mathrm{I}_{\mathbb{Z}}(\mathscr{P}). \quad \blacksquare$$

LEMMA 2. *If* $\beta - \alpha \in \mathrm{I}_{\mathbb{Q}}(\mathscr{P})$, *then* $\alpha \equiv \beta(\mathscr{P})$. *In particular, if* $\beta - \alpha = \sum_{i=1}^{w} (\beta_i - \alpha_i)\, g_i$ *for* $g_i \in \mathbb{Q}[X]$, *then there is a derivation* $\alpha = \gamma_0 \to \gamma_1 \to \cdots \to \gamma_n = \beta$ *of* β *from* α *in* $\mathscr{P}$, *such that for* $j \in I_n$,

$$\mathrm{length}(\gamma_j) \leqslant \max\{\deg(\beta_i\, g_i);\ i \in I_w\}.$$

Proof. Let $d \in \mathbb{N}$ be a common denominator for all the rational coefficients in the g_i, $i \in I_w$. Then we may assume without loss of generality that $\beta \neq \alpha$ and

$$d\beta - d\alpha = \sum_{m=1}^{n} (\beta_{i_m} - \alpha_{i_m})\, g'_m \qquad \text{for some} \quad n \geqslant 1,$$

where the $g'_m \in \mathbb{Z}[X]$, $m \in I_n$, are all monomials with coefficient $+1$, and $\deg(g'_m) \leqslant \deg(g_{i_m})$ for $m \in I_n$.

As α appears as a term on the left side of this polynomial identity and $\alpha \neq \beta$ there must be some $r \in I_n$ such that $\alpha = \alpha_{i_r} g'_r$, implying

(i) $d\beta - (d-1)\,\alpha - \beta_{i_r} g'_r = \sum_{m \in I_n - \{r\}} (\beta_{i_m} - \alpha_{i_m})\, g'_m$,

(ii) $\alpha \to \beta_{i_r} g'_r(\mathscr{P})$.

If $\beta_{i_r} g'_r = \beta$ we are finished, otherwise we may repeat the above argument for $\beta_{i_r} g'_r$ in place of α, and by induction on n obtain a derivation

$$\alpha \to \gamma_1 \to \cdots \to \gamma_{n'} = \beta \qquad \text{with} \quad n' \leqslant n,$$

where each γ_k is of the form $\beta_{i_r} g'_r$ for some $r = r(k) \in I_n$. $\blacksquare$

Note that the above mapping from the *CSG* problem instance $(\alpha, \beta, \mathscr{P})$ to the *PI* problem instance $\langle \beta - \alpha,\ \beta_1 - \alpha_1, \ldots,\ \beta_w - \alpha_w \rangle$ is computationally trivial and size preserving, so Lemmas 1 and 2 imply that *CSG* is log-lin reducible to *PI*.

From the work in [15],[2] we can derive the following:

PROPOSITION. *Let* $X=\{x_1,\ldots,x_v\}$; $p, p_1,\ldots,p_w \in \mathbb{Q}[X]$; *and* $d =_{\text{def}} \max\{\deg(p_i);\ i\in I_w\}$. *If* $p\in(p_1,\ldots,p_w)$, *then there exist* $g_1,\ldots,g_w\in\mathbb{Q}[X]$ *such that*

(i) $p=\sum_{i=1}^{w} p_i g_i$;

(ii) $(\forall\, i\in I_w)[\deg(g_i)\leqslant \deg(p)+(wd)^{2^v}]$.

Proof. For the reader's convenience an improved proof of this Proposition is given in the Appendix. ∎

We should like to mention that the general problem of the solvability of linear equations over $R[X]$ for rings R other than $\mathbb{Q}$ or $\mathbb{Z}$ has been investigated in [36, 38]. We also note that Lemmas 1 and 2 imply that $\beta-\alpha\in \mathrm{I}_{\mathbb{Q}}(\mathscr{P})$ iff $\beta-\alpha\in\mathrm{I}_{\mathbb{Z}}(\mathscr{P})$. This condition does not hold for ideals generated by *arbitrary* polynomials in $\mathbb{Z}[X]$.

4. An Exponential Space Upper Bound

The above Proposition and the lemmas of the previous section easily yield:

LEMMA 3. *Let* $S=\{s_1,\ldots,s_v\}$ *and* $\mathscr{P}=\{\alpha_i\equiv\beta_i;\ i\in I_w\}$ *be a commutative semigroup presentation over S. Then, for* $\alpha,\beta\in S^*$, $\alpha\equiv\beta(\mathscr{P})$ *iff there is a derivation* $\alpha=\gamma_0\to\gamma_1\to\cdots\to\gamma_n=\beta(\mathscr{P})$ *of* β *from* α *such that*

$$\text{length}(\gamma_i)\leqslant 2^{2^{c\cdot \text{size}(\alpha,\beta,\mathscr{P})}} \quad \textit{for all} \quad i\in\{0,\ldots,n\},$$

where $c>0$ *is some universal constant independent of* $(\alpha,\beta,\mathscr{P})$.

Proof. Note that $\deg(\beta-\alpha)$ and $\deg(\beta_i-\alpha_i)$, for $i\in I_w$, are all bounded by $2^{\text{size}(\alpha,\beta,\mathscr{P})}$. Further, $\text{size}(\alpha,\beta,\mathscr{P})$ is also an upper bound on the number w of generators of the polynomial ideal $\mathrm{I}_{\mathbb{Q}}(\mathscr{P})$. Thus the upper bound of the lemma follows from Lemma 2 and part (ii) of the above Proposition. ∎

Hence we conclude:

THEOREM 1. *There is a (deterministic) Turing machine M and some constant* $d>0$, *such that for any instance* $(\alpha,\beta,\mathscr{P})$, *M decides whether* $\alpha\equiv\beta(\mathscr{P})$ *using at most space* $2^{d\cdot\text{size}(\alpha,\beta,\mathscr{P})}$.

Proof sketch. A nondeterministic Turing machine may determine whether $\alpha\equiv\beta(\mathscr{P})$ by generating a derivation $\alpha=\gamma_0\to\gamma_1\to\cdots\to\gamma_n=\beta$ iff

[2] Some parts of [15] which we do not use have been improved in [38].

there is one. For this purpose, obviously only two consecutive words γ_{i-1} and γ_i in the derivation have to be kept in storage at any time in order to check whether $\gamma_{i-1} \to \gamma_i(\mathscr{P})$. Clearly, the words γ_i can be represented by writing down a representation of $\Phi(\gamma_i)$, with numbers in radix notation. This representation therefore requires only $O(\log(\text{length}(\gamma_i)))$ tape squares. By Lemma 3, there is some universal constant $d' > 0$ such that this nondeterministic Turing machine needs at most $2^{d' \cdot \text{size}(\alpha, \beta, \mathscr{P})}$ tape cells on an instance $(\alpha, \beta, \mathscr{P})$ in order to determine whether $\alpha \equiv \beta(\mathscr{P})$.

Nondeterministic Turing machines can be simulated by ordinary deterministic ones for which the number of required tape cells at most gets squared [37]. ∎

5. Semigroup Presentations and Bounded Counter Machines

An *n-counter machine* models a computer having n registers each of which may hold an arbitrary integer. All registers initially contain 0. The machine can, in one atomic operation, modify any one of its registers by adding -1, 0, or 1 to its current value, or test whether a specified register contains 0 and branch on the outcome of this test. In the sequel, it suffices to consider 3-counter machines. A 3-counter machine can be used to compute any partial recursive function [30].

Formally, a 3-counter machine C consists of a finite set Q of states, a pair of distinguished states q_0 and $q_a \in Q$ (where q_0 is called the *initial* and q_a the *accepting* state), and a (transition) function

$$\delta\colon (Q - \{q_a\}) \to (Q \times \{0, \pm 1\} \times I_3) \cup (Q \times Q \times I_3).$$

The *computation* of C is given by the (possibly infinite) sequence $c^0, c^1, \ldots$ of *instantaneous descriptions* $c^i \in Q \times \mathbb{Z}^3$, where

(i) $c^0 = (q_0, 0, 0, 0)$, and

(ii) if $i \in \mathbb{N}$, $c^i = (q, z_1, z_2, z_3)$ with $q \neq q_a$, and

(α) $\delta(q) = (q', d, k) \in Q \times \{0, \pm 1\} \times I_3$, then

$$c^{i+1} = (q', z_1', z_2', z_3'), \qquad \text{where} \quad z_i' = \begin{cases} z_i + d & \text{if} \quad i = k \\ z_i & \text{otherwise.} \end{cases}$$

(β) $\delta(q) = (q', q'', k) \in Q \times Q \times I_3$, then

$$\begin{aligned} c^{i+1} &= (q', z_1, z_2, z_3) && \text{if} \quad z_k = 0; \\ &= (q'', z_1, z_2, z_3) && \text{otherwise.} \end{aligned}$$

c^i_{1+k} is referred to as the contents of the kth *counter* after i steps, for $k \in I_3$.

C is said to *terminate with empty counters* iff its computation contains the quadruple $(q_a, 0, 0, 0)$. Note that $(q_a, 0, 0, 0)$ then is the last element in the computation of C. As we will only be concerned with termination with empty counters, we will for convenience henceforth refer simply to *termination*.

We define the *size* of C to be the cardinality of its state set Q.

Now let $n \in \mathbb{N}$. The computation of some 3-counter machine C is said to be *bounded* by n iff after any step in the computation the contents of all three counters are $\geqslant 0$ and $\leqslant n$.

From the results in [10], one can easily derive (and we state without proof):

LEMMA 4. *The set*

$$ESC \stackrel{\text{def}}{=} \{C;\ C \textit{ is a terminating 3-counter machine whose computation is bounded by } 2^{2^{\text{size}(C)}}\},$$

is exponential space complete under log-lin reducibility.

Henceforth, we shall refer to this *e*xponential *s*pace *c*omplete problem as ESC.

ESC will be used to prove an exponential space lower bound for CSG. We shall, in the remaining part of this section and in the next, show how to construct from any given 3-counter machine C a commutative semigroup presentation of size $O(\text{size}(C))$ such that ESC reduces to CSG. In this section, we shall finitely present a commutative semigroup to which ESC can be reduced. However, the presentation $\mathscr{P}'_C$ we describe will still be too big. In the next section then, we shall show how to embed the relevant part of this semigroup into one given by a small presentation $\mathscr{P}_C$.

Henceforth, let $e_n =_{\text{def}} 2^{2^n}$.

The most straightforward way to represent a configuration $c = (q, z_1, z_2, z_3)$ of a 3-counter machine C would be by a word of the form

$$qh_1^{z_1}h_2^{z_2}h_3^{z_3},$$

where h_1, h_2, h_3 are distinct symbols.

However, we have no direct way to "simulate" the zero-test capability of C with this representation. But as the counters of all $C \in ESC$ are always bounded by e_n (where $n =_{\text{def}} \text{size}(C)$), we can choose a variant representation in which each element (q, z_1, z_2, z_3) in the computation of $C = (Q, \delta)$ is represented by a word $w(q, z_1, z_2, z_3) \in \bar{Q}^*$, where $\bar{Q}$ is the disjoint union $Q \uplus \{g_1, h_1, g_2, h_2, g_3, h_3\}$:

$$w(q, z_1, z_2, z_3) \stackrel{\text{def}}{=} qg_1^{e_n - z_1}h_1^{z_1}g_2^{e_n - z_2}h_2^{z_2}g_3^{e_n - z_3}h_3^{z_3}.$$

Henceforth let $C=(Q,\delta)$ be some fixed 3-counter machine of size n. We define the commutative semigroup presentation $\mathscr{P}'_C$ over the alphabet $\bar{Q}$ to contain exactly the following equivalences:

For every $q \in Q$ with $\delta(q)=(q',d,k) \in Q \times \{0,\pm 1\} \times I_3$:

$$q \equiv q' \qquad \text{if} \quad d=0, \tag{A}$$

$$qg_k \equiv q'h_k \qquad \text{if} \quad d=1, \tag{B}$$

and

$$qh_k \equiv q'g_k \qquad \text{if} \quad d=-1. \tag{C}$$

For every $q \in Q$ with $\delta(q)=(q',q'',k) \in Q \times Q \times I_3$:

$$qh_k \equiv q''h_k, \tag{D}$$

and

$$qg_k^{e_n} \equiv q'g_k^{e_n}. \tag{E}$$

Also, let

$$W \stackrel{\text{def}}{=} \{w(q,z_1,z_2,z_3);\, q \in Q, \text{ and } 0 \leqslant z_1, z_2, z_3 \leqslant e_n\}.$$

If $\alpha \in W$, then by definition, $\Phi(\alpha, g_k)+\Phi(\alpha,h_k)=e_n$, for $k \in I_3$. Moreover, if $\beta \equiv \alpha(\mathscr{P}'_C)$, a simple induction on the length of a derivation of β from α in $\mathscr{P}'_C$ shows that also $\beta \in W$, and hence that, in particular, $\Phi(\beta, g_k)+\Phi(\beta,h_k)=e_n$, where $k \in I_3$. This invariance is the reason that use of equivalence (E) in a derivation corresponds to a branch-on-zero step in the computation of C.

It follows that $C \in ESC$ implies $w(q_0,0,0,0) \equiv w(q_a,0,0,0)(\mathscr{P}'_C)$, because the computation $c^0, c^1, \ldots$ of C is simulated, in a step by step fashion, by a corresponding derivation $w(c^0) \to w(c^1) \to \cdots$, using the above equivalences (A)–(E) only as semi-Thue productions from left to right.

It also turns out that the same line of argument as in [34] for the case of noncommutative semigroups provides the converse implication, yielding:

LEMMA 5. $w(q_0,0,0,0) \equiv w(q_a,0,0,0)(\mathscr{P}'_C) \Leftrightarrow C \in ESC$.

For a detailed proof see [5].

6. Succinct Semigroup Presentations

For $n \in \mathbb{N}$ we construct a commutative semigroup presentation $\mathscr{P}_n$ of size $O(n)$ containing generators S, F and B such that, in essence, FB^{e_n} is the only word containing F that is derivable from S in $\mathscr{P}_n$. The presentation $\mathscr{P}_n$ and its set G_n of generators are defined by induction on n, noting the fact that $e_{n+1} = (e_n)^2$. For technical reasons, $\mathscr{P}_n$ will contain four different symbols $B_1, \ldots, B_4$ each acting like the B above. Let

$$G_0 \stackrel{\text{def}}{=} \{s, f, c_1, c_2, c_3, c_4, b_1, b_2, b_3, b_4\} \quad \text{and} \quad \mathscr{P}_0 \stackrel{\text{def}}{=} \{sc_i \equiv fc_i b_i^2 \,;\, i \in I_4\}.$$

For $m > 0$, let $\{S, Q_1, Q_2, Q_3, Q_4, F, C_1, C_2, C_3, C_4, B_1, B_2, B_3, B_4\}$ be distinct symbols not in G_{m-1}. Then

$$G_m \stackrel{\text{def}}{=} G_{m-1} \uplus \{S, Q_1, Q_2, Q_3, Q_4, F, C_1, C_2, C_3, C_4, B_1, B_2, B_3, B_4\}.$$

The elements of G_0 are of *level* 0, and for $m > 0$, the elements of $G_m - G_{m-1}$ are of *level m*. For notational convenience, we now let the upper case letters $S_1, \ldots, B_4$ denote the generators of level n ($n > 0$), and let the lower case letters $s, \ldots, b_4$ denote the corresponding generators of level $n-1$. $\mathscr{P}_n$ then is the union of $\mathscr{P}_{n-1}$ and the following equivalences:

$$S \equiv Q_1 sc_1, \tag{a}$$

$$Q_1 fc_1 b_1 \equiv Q_2 sc_2, \tag{b}$$

$$Q_2 fc_2 \equiv Q_3 fc_3, \tag{c}$$

$$Q_3 sc_3 b_1 \equiv Q_2 sc_2 b_4, \tag{d}$$

$$Q_3 sc_3 \equiv Q_4 fc_4 b_4, \tag{c}$$

$$Q_4 sc_4 \equiv F, \tag{f}$$

and, for $i \in I_4$,

$$Q_2 C_i f b_2 \equiv Q_2 C_i B_i f b_3. \tag{g)–(i}$$

LEMMA 6. *Let S, F, C_i, B_i, for $i \in I_4$ be of level n. Then*

$$SC_i \equiv FC_i B_i^{e_n}(\mathscr{P}_n) \qquad \textit{for} \quad i \in I_4.$$

Proof. The proof is done by induction on n.
For $n = 0$, $\mathscr{P}_0$ contains exactly the equivalences claimed.
For $n > 0$, we have for $i \in I_4$

$SC_i \equiv C_i Q_1 sc_1$	by	(a)
$\equiv C_i Q_1 fc_1 b_1^{e_{n-1}}$	by	induction hypothesis
$\equiv C_i b_1^{e_{n-1}-1} Q_2 sc_2$	by	(b)
$\equiv C_i b_1^{e_{n-1}-1} Q_2 fc_2 b_2^{e_{n-1}}$	by	induction hypothesis

$$\begin{array}{lll}
\equiv C_i b_1^{e_{n-1}-1} Q_2 f c_2 b_3^{e_{n-1}} B_i^{e_{n-1}} & \text{by} & \text{(g)–(j)} \\
\equiv C_i B_i^{e_{n-1}} b_1^{e_{n-1}-1} Q_3 f c_3 b_3^{e_{n-1}} & \text{by} & \text{(c)} \\
\equiv C_i B_i^{e_{n-1}} b_1^{r_{n-1}-1} Q_3 s c_3 & \text{by} & \text{induction hypothesis} \\
\equiv C_i B_i^{e_{n-1}} b_1^{e_{n-1}-2} b_4 Q_2 s c_2 & \text{by} & \text{(d)} \\
\equiv \cdots \equiv C_i B_i^{e_{n-1} e_{n-1}} b_4^{e_{n-1}-1} Q_3 s c_3 & \text{by} & \text{iteration of the previous five lines} \\
\equiv C_i B_i^{e_n} Q_4 f c_4 b_4^{e_{n-1}} & \text{by} & \text{(e)} \\
\equiv C_i B_i^{e_n} Q_4 s c_4 & \text{by} & \text{induction hypothesis} \\
\equiv F C_i B_i^{e_n} & \text{by} & \text{(f).} \quad \blacksquare
\end{array}$$

We are now going to show that the derivation given in the proof of the previous lemma is the only repetition-free derivation from SC_i in $\mathscr{P}_n$ that produces a word containing the level n symbol F. For this purpose, we first establish some technical properties of derivations in $\mathscr{P}_n$.

Let S, C_1 be of level n, and $\alpha \in G_n^*$ such that $\alpha \equiv SC_1(\mathscr{P}_n)$. Define the *height* $h(\alpha)$ by

$$h(\alpha) \stackrel{\text{def}}{=} \min\{m \in \mathbb{N};\ \Phi(\alpha, c_i) > 0, \text{ for some } c_i \text{ of level } m\}.$$

Then we have:

LEMMA 7. *Let* $SC_1 = \gamma_0 \to \gamma_1 \to \cdots \to \gamma_r = \alpha(\mathscr{P}_n)$ *be a derivation of* α *from* SC_1 *in* $\mathscr{P}_n$. *Then*:

(ii) $\displaystyle\sum_{i=1}^{4} \Phi(\alpha, c_i) = 1$ *if* $c_1, \ldots, c_4$ *are of level* m *with* $h(\alpha) \leqslant m \leqslant n$,
$= 0$ *otherwise*;

(ii) $\displaystyle\sum_{i=1}^{4} \Phi(\alpha, q_i) = 1$ *if* $q_1, \ldots, q_4$ *are of level* m *with* $h(\alpha) < m \leqslant n$,
$= 0$ *otherwise*;

(iii) $\Phi(\alpha, s) + \Phi(\alpha, f) = 1$ *if* s, f *are of level* $h(\alpha)$,
$= 0$ *otherwise*;

(iv) $|h(\gamma_i) - h(\gamma_{i-1})| \leqslant 1$ *for all* $i \in I_r$;

(v) *only equivalences in* $\mathscr{P}_{h(\alpha)+1} - \mathscr{P}_{h(\alpha)-1}$ *are applicable to* α (*here*, $\mathscr{P}_{-1}$ *is taken to be* $\varnothing$); *the height decreases iff an equivalence in* $\mathscr{P}_{h(\alpha)}$ *is applied.*

Proof. The proof is by induction on the length r of the derivation. The details are left to the reader. ∎

LEMMA 8. *Let S, F, C_i, B_i, $i \in I_4$, be of level n, and let $\alpha \in FG_n^*$. If $SC_i \equiv \alpha(\mathscr{P}_n)$ and α contains an occurrence of S or F, then either $\alpha = SC_i$ or $\alpha = FC_iB_i^{e_n}$.*

Proof. The conclusion of the lemma is immediate for $n=0$. Hence, assume $n>0$, and let

$$SC_1 = \gamma_0 \to \gamma_1 \to \cdots \to \gamma_r = \alpha(\mathscr{P}_n) \tag{$*$}$$

be any *repetition-free* derivation of α in $\mathscr{P}_n$. We prove by induction on n that $(*)$ must be the derivation given in the proof of Lemma 6.

First, note that because α contains S or F, Lemma 7(iii) implies $h(\alpha)=n$. Second, note that besides SC_1 and α there is no other word γ_i in $(*)$ of height n: if $h(\gamma_i)=n$ for some minimal $0<i<r$ then γ_i would contain S or F by Lemma 7(iii). As S and F appear only in equivalences (a) and (f), respectively, of $\mathscr{P}_n$, inspection of $\mathscr{P}_n$ shows that only the reversal of the equivalence used from γ_{i-1} to γ_i would be applicable to γ_i, causing the repetition $\gamma_{i+1}=\gamma_{i-1}$.

As only equivalence (a) of levels n and $n-1$ is applicable to γ_0 and γ_1, respectively, we have

$$\gamma_1 = C_1Q_1sc_1, \qquad h(\gamma_1)=n-1, \qquad h(\gamma_2)=n-2.$$

Because of Lemma 7(iv) there must be a first word γ_{i_1} in $(*)$ after γ_1 which also has height $n-1$. Hence, by Lemma 7(v), only equivalences in $\mathscr{P}_{n-1}$ could have been used in the subderivation

$$\gamma_1 \to \cdots \to \gamma_{i_1}(\mathscr{P}_n).$$

As these equivalences do not contain any occurrences of symbols of level n we may rewrite γ_i in the above subderivation as $Q_1C_1\gamma_i'$, where $\gamma_i' \in G_{n-1}^*$. Thus, $\gamma_1' = sc_1$, and by Lemma 7(iii), γ_{i_1}' contains either s or f. Hence,

$$sc_1 = \gamma_1' \to \cdots \to \gamma_{i_1}'(\mathscr{P}_{n-1}).$$

We conclude from the induction hypothesis for $n-1$ that

$$\gamma_{i_1}' = fc_1b_1^{e_{n-1}} \qquad \text{and} \qquad \gamma_{i_1} = Q_1C_1fc_1b_1^{e_{n-1}}.$$

Because $(*)$ is repetition-free, the only equivalence now applicable is (b) of level n, i.e.,

$$\gamma_{i_1+1} = Q_2C_1b_1^{e_{n-1}-1}sc_2, \qquad h(\gamma_{i_1+1})=n-1, \qquad h(\gamma_{i_1+2})=n-2.$$

Again, let γ_{i_2} be the firstword in (∗) after γ_{i_1} of height $n-1$. As above, only equivalences in $\mathscr{P}_{n-1}$ can be used between γ_{i_1+1} and γ_{i_2}. What is more, c_2 occurs in γ_i for all i with $i_1 < i \leqslant i_2$ as only equivalences of level n can possibly change c_2 to some other c_k. Thus by Lemma 7(i), equivalence (g) of level $n-1$ cannot be applied to any γ_i with $i_1 < i \leqslant i_2$. Therefore, we can once more rewrite γ_i as $Q_2 C_1 b_1^{e_{n-1}-1}\gamma_i'$ with $\gamma_i' \in G_{n-1}^*$ for $i_1 < i \leqslant i_2$, and have

$$sc_2 = \gamma_{i_1+1}' \to \cdots \to \gamma_{i_2}'(\mathscr{P}_{n-1}).$$

By Lemma 7(iii), γ_{i_2}' contains either s or f, and hence the induction hypothesis implies

$$\gamma_{i_2} = Q_2 C_1 b_1^{e_{n-1}-1} f c_2 b_2^{e_{n-1}}.$$

Now only equivalence (g) of level n can be applied, say k times, for any $0 \leqslant k \leqslant e_{n-1}$, and then equivalence (c), producing

$$\gamma_{i_3} = Q_3 C_1 B_1^k b_2^{e_{n-1}-1} f c_3 b_3^k.$$

The only rule now applicable without causing repetition is (f) of level $n-1$ so that $h(\gamma_{i_3+1}) = n-2$. Let $i_4 > i_3$ be minimal such that $h(\gamma_{i_4}) = n-1$. Note that i_4 must exist because of Lemma 7(iv). Also note that now between γ_{i_3} and γ_{i_4}, c_3 is present in all words. Therefore, the equivalences (g) and (h) of level $n-1$ are not applicable. We may thus, as above, parse γ_i as $Q_3 C_1 B_1^k b_1^{e_{n-1}-1} b_2^{e_{n-1}-k} \gamma_i'$ with $\gamma_i' \in G_{n-1}^*$ for $i_3 \leqslant i \leqslant i_4$, and obtain

$$fc_3 b_3^k = \gamma_{i_3}' \to \cdots \to \gamma_{i_4}'(\mathscr{P}_{n-1})$$

where either s or f occurs in γ_{i_4}'.

Assume first that $\gamma_{i_4}' = fc_3\eta$ with $\eta \in G_{n-1}^*$, and, of course, $\eta \neq b_3^k$. Then, by Lemma 6 there is a derivation

$$sc_3 \to \cdots \to fc_3 b_3^k b_3^{e_{n-1}-k} \to \cdots \to fc_3 b_3^{e_{n-1}-k}\gamma(\mathscr{P}_{n-1}).$$

As $b_3^{e_{n-1}-k} \neq b_3^{e_{n-1}}$, this contradicts the induction hypothesis.

Otherwise, if $\gamma_{i_4}' = sc_3\eta$ with $\eta \in G_{n-1}^*$, note that $\mathscr{P}_{n-1}$ is symmetric and consider the derivation

$$\begin{aligned} sc_3 &\to \cdots \to fc_3 b_3^k b_3^{e_{n-1}-k} && \text{(by Lemma 6)} \\ &\to \cdots \to sc_3 \eta b_3^{e_{n-1}-k} && \text{(by assumption).} \end{aligned}$$

But the induction hypothesis for $n-1$ implies that $sc_3 = sc_3 \eta b_3^{e_{n-1}-k}$. In other words, η is the empty word and $k = e_{n-1}$, so that

$$\gamma_{i_4} = Q_3 C_1 B_1^{e_{n-1}} b_1^{e_{n-1}-1} sc_3.$$

Now only either equivalence (d) or (c) of level n can be applied. As there has to be another successor of height $n-1$, equivalence (e) is excluded here as can be seen by an argument analogous to the one just given for $\gamma'_{i_3} = fc_3 b_3^k$. Hence,

$$\gamma_{i_4+1} = Q_2 C_1 B_1^{e_{n-1}} b_1^{e_{n-1}-2} b_4 s c_2 \qquad \text{by equivalence (d).}$$

Similarly, if now (b) (from right to left) were applied, the induction hypothesis would imply that there is no $i > i_4 + 1$ with $h(\gamma_i) = n-1$ which is impossible. Hence, we are forced to apply to γ_{i_4+1} equivalence (a) of level $n-1$, and therefore obtain $h(\gamma_{i_4+2}) = n-2$. We may now iterate $e_{n-1} - 1$ times the argument which has been used for the subsequence $\gamma_{i_1} \to \cdots \to \gamma_{i_4}(\mathscr{P}_n)$, and thus obtain some $i_5 > i_4 + 2$ such that

$$\gamma_{i_5} = Q_3 C_1 B_1^{e_{n-1} e_{n-1}} s c_3 b_4^{e_{n-1}-1}.$$

Here only (e) is applicable:

$$\gamma_{i_5+1} = Q_4 C_1 B_1^{e_n} f c_4 b_4^{e_{n-1}}.$$

Because only equivalence (f) of level $n-1$ may be used we obtain $h(\gamma_{i_5+2}) = n-2$. By Lemma 7(iv) there has to be a minimal $i_6 > i_5 + 1$ with $h(\gamma_{i_6}) = n-1$, so we can, as above, conclude from the induction hypothesis that

$$\gamma_{i_6} = Q_4 C_1 B_1^{e_n} s c_4,$$

and obviously,

$$\gamma_{i_6+1} = \gamma_r = F C_1 B_1^{e_n} \qquad \text{by equivalence (f).}$$

With C_1 replaced by C_2, C_3, or C_4, the proof runs analogously. ∎

7. An Exponential Space Lower Bound for *CSG*

Given some 3-counter machine $C = (Q, \delta)$ of side n, the commutative semigroup presentation $\mathscr{P}_C$ is constructed as follows:

Assume without loss of generality that equivalences (A)–(D) of $\mathscr{P}'_C$ are over an alphabet disjoint from the alphabet G_n of $\mathscr{P}_n$, with the exception that the symbol g_k in (A)–(D) is taken to be $B_k \in G_n$ for all $k \in I_3$. Then $\mathscr{P}_C$ is defined to contain $\mathscr{P}_n$ and all equivalences (A)–(D) of $\mathscr{P}'_C$.

For every equivalence $qg_k^{e_n} \equiv q'g_k^{e_n}$ of form (E) in $\mathscr{P}_C'$, i.e., for every test state $q \in Q$, let q_r, q_e be two new symbols $\notin \bar{Q} \cup G_n$. Then, instead of the equivalence $qg_k^{e_n} \equiv q'g_k^{e_n}$ in $\mathscr{P}_C'$, the following equivalences are also added to $\mathscr{P}_C$ (where $S, F, C_1, C_2, C_3 \in G_n$ are of level n):

$$q \equiv q_r F C_k, \tag{k}$$

$$q_r S C_k \equiv q_c S C_k, \tag{l}$$

$$q_c F C_k \equiv q'. \tag{m}$$

Finally, let

$$q_{01} \equiv q_{02} S C_1, \tag{n}$$

$$q_{02} F C_1 \equiv q_{03} S C_2, \tag{o}$$

$$q_{03} F C_2 \equiv q_{04} S C_3, \tag{p}$$

$$q_{04} F C_3 \equiv q_0, \tag{q}$$

$$q_a \equiv q_{a4} F C_3, \tag{r}$$

$$q_{a4} S C_3 \equiv q_{a3} F C_2, \tag{s}$$

$$q_{a3} S C_2 \equiv q_{a2} F C_1, \tag{t}$$

$$q_{a2} S C_1 \equiv q_{a1} \tag{u}$$

be in $\mathscr{P}_C$. These are all the equivalences in $\mathscr{P}_C$.

The auxiliary symbols q_{02}, q_{03}, q_{04} are used to expand q_{01} into the actual representation $w(q_0, 0, 0, 0)$ of the initial instantaneous description. Similarly, the final configuration $w(q_a, 0, 0, 0)$ is reduced to q_{a1} using the auxiliary symbols q_{a4}, q_{a3}, q_{a2}.

Remember that $W = \{w(q, z_1, z_2, z_3);\ q \in Q,\ \text{and}\ 0 \leqslant z_1, z_2, z_3 \leqslant e_n\}$. Define further $\mathscr{W}$ to be the subset of the commutative semigroup presented by $\mathscr{P}_C'$ which is given by the words in W. Then we have:

LEMMA 9. *There is a semigroup homomorphism from the commutative semigroup presented by $\mathscr{P}_C'$ into the one presented by $\mathscr{P}_C$ which is faithful (i.e., injective) on $\mathscr{W}$.*

Proof. Let ι map g_k to B_k for $k \in I_3$ and be the identity mapping on $\bar{Q}$ otherwise. We claim that ι provides an embedding of $\mathscr{W}$, namely, for $w, w' \in W$,

$$w \equiv w'(\mathscr{P}_C') \Leftrightarrow \iota(w) \equiv \iota(w')(\mathscr{P}_C).$$

Any form (E) equivalence $qg_k^{e_n} \equiv q'g_k^{e_n}$ of $\mathscr{P}_C'$ yields a corresponding equivalence in $\mathscr{P}_C$ because

$$\begin{aligned} \iota(qg_k^{e_n}) = qB_k^{e_n} &\equiv q_rFC_kB_k^{e_n} && \text{by} \quad (\mathrm{k}) \\ &\equiv q_rSC_k && \text{by} \quad \text{Lemma 6} \\ &\equiv q_eSC_k && \text{by} \quad (\mathrm{l}) \\ &\equiv q_eFC_kB_k^{e_n} && \text{by} \quad \text{Lemma 6} \\ &\equiv q'B_k^{e_n} = \iota(q'g_k^{e_n}) && \text{by} \quad (\mathrm{m}). \end{aligned}$$

Since equivalences (A)–(D) of $\mathscr{P}_C'$ are also in $\mathscr{P}_C$, we conclude that

$$w \equiv w'(\mathscr{P}_C') \Rightarrow \iota(w) \equiv \iota(w')(\mathscr{P}_C) \qquad \text{for all} \quad w, w' \in \bar{Q}^*.$$

To prove the converse implication, suppose

$$\alpha = \iota(w(q, z_1, z_2, z_3)) \equiv \iota(w(q', z_1', z_2', z_3')) = \beta(\mathscr{P}_C),$$

and let $\alpha = \gamma_0 \to \gamma_1 \to \cdots \to \gamma_r = \beta(\mathscr{P}_C)$ be a repetition-free derivation. Let $\gamma_m \to \gamma_{m+1}$ be the first step in this derivation that uses one of the equivalences (k)–(u). Then clearly $\iota^{-1}(\gamma_m) \equiv \iota^{-1}(\alpha)(\mathscr{P}_C')$ as W is closed under the $\mathscr{P}_C'$ derivation rules. There are the following four possibilities:

(i) If the equivalence used on γ_m is (q), it follows from Lemma 8 that only equivalences in $\mathscr{P}_n$ and (p), (o), (n) can be used, i.e., the only words of height n derivable from γ_{m+1} are those obtained from γ_{m+1} by successively replacing

$$\begin{array}{llllll} q_{04}FC_3B_3^{e_n} & \text{by} & q_{04}SC_3 & \text{and} & q_{03}FC_2, & \text{then} \\ q_{03}FC_2B_2^{e_n} & \text{by} & q_{03}SC_2 & \text{and} & q_{02}FC_1, & \text{and finally} \\ q_{02}FC_1B_1^{e_n} & \text{by} & q_{02}SC_1 & \text{and} & q_{01}. & \end{array}$$

Hence, in order to reach β the above derivation cannot be repetition-free. So equivalence (q) cannot be used.

(ii) A similar argument eliminates the possibility of using equivalence (u).

(iii) If equivalence (k) is used, again only equivalences in $\mathscr{P}_n$ can be applied thereafter. Lemma 8 then implies that the next two words of height n in the above derivation are those obtained from γ_m by replacing $qB_k^{e_n}$ by q_rSC_k and q_eSC_k, where k is determined by $\delta(q)$. The next word of height n

is then obtained by substituting $q_e F C_k B_k^{e_n}$ for $q_e S C_k$. Now only equivalence (m) is applicable, and hence there is some $m' > m$ such that

$$\gamma_{m'} \in W \qquad \text{and} \qquad \iota^{-1}(\gamma_{m'}) \equiv \iota^{-1}(\gamma_m)(\mathscr{P}'_C).$$

(iv) If equivalence (m) is used an analogous argument applies.

By induction on the length of derivations over $\mathscr{P}_C$, we may assume that $\iota^{-1}(\gamma_{m'}) \equiv \iota^{-1}(\beta)(\mathscr{P}'_C)$. Therefore, $\alpha \overset{*}{\to} \beta(\mathscr{P}_C)$ implies that

$$\iota^{-1}(\alpha) \equiv \iota^{-1}(\beta)(\mathscr{P}'_C),$$

which concludes the proof. ∎

LEMMA 10. *Let C be a 3-counter machine and $\mathscr{P}_C$ the finite commutative semigroup presentation constructed above. Then*

$$C \in ESC \Leftrightarrow q_{01} \equiv q_{a1}(\mathscr{P}_C).$$

Proof. By Lemmas 5 and 9

$$C \in ESC \Leftrightarrow \iota(w(q_0, 0, 0, 0)) \equiv \iota(w(q_a, 0, 0, 0))(\mathscr{P}_C).$$

But by the same argument used for case (i) in the proof of Lemma 9, we conclude that

$$q_{01} \equiv q_{a1}(\mathscr{P}_C) \Leftrightarrow \iota(w(q_0, 0, 0, 0)) \equiv \iota(w(q_a, 0, 0, 0))(\mathscr{P}_C). \quad ∎$$

THEOREM 2. *CSG is exponential space complete with respect to log-lin reducibility.*

Proof. Lemma 10 shows that ESC reduces to CSG. It follows from the construction of $\mathscr{P}_C$ that $\text{size}(\mathscr{P}_C) = O(\text{size}(C))$. The reader can verify that the construction of $(q_{01}, q_{a1}, \mathscr{P}_C)$ from C can in fact be carried out in logarithmic space. Hence the reduction is log-lin. ∎

Part (b) of the Main Theorem stated in the Introduction is an immediate corollary of Theorem 2 and the properties of log-lin reductions described in Section 2. Since CSG is log-lin reducible to PI, we also obtain:

COROLLARY. *The membership problem for polynomial ideals PI is exponential space hard.*

Exponential space completeness yields other interesting consequences in addition to the preceding corollary. For example, not only does every decision procedure for CSG require exponential space but there is no optimally efficient procedure: given any procedure for CSG one can find another procedure which used no more space on any problem instance but which uses a bounded amount of space on an infinite set of problem instances for which the original procedure required exponentially growing space. (This property is known as *effective infinitely often speedup*, cf. [4, 41].)

We repeat our earlier remark that Theorem 2 and consequently part (b) of the Main Theorem, also hold if we allow only unary notation for representing instances of *CSG*. The reason is that the values of the exponents needed for the presentation $\mathscr{P}_C$ are at most two.

An overview of the preceding argument reveals that the complexity of the uniform word problem for commutative semigroups depends on how rapidly growing a function f: $\mathbb{N} \to \mathbb{N}$ may be such that one can express equations of the form

$$s_1 \equiv s_2^{f(n)}$$

with presentations of size $O(n)$. The space requirements of the corresponding word problem will grow at least proportionally to $\log(f)$. The results of Section 6 show that $f(n) = 2^{2^n}$ is possible using either unary or exponential notation for presentations, and the results of Section 4 imply that f cannot grow more than double exponentially using these notations. However, if we liberalize notational conventions further, for example, allowing iterated exponential notation such as

$$s_1 \equiv s_2^{2^{2^{\cdot^{\cdot^{\cdot^2}}}}},$$

then $f(n)$ may be as large as $2^{2^{\cdot^{\cdot^{\cdot^2}}}}$ with exponentials up to height n. This version of the uniform word problem for commutative semigroups (using iterated exponential notation) therefore cannot be decided in space bounded by any finite composition of exponentials. In general, by introducing successively more powerful abbreviations, we can obtain arbitrarily complex decidable variations of *CSG*.

8. Conclusion and Open Problems

Theorems establishing the degree of unsolvability of decision problems have become familiar in most areas of mathematics during the past 50 years. The same philosophical and practical issues which have motivated the analysis of degrees of unsolvability serve equally to motivate the analysis of degrees of solvability, i.e., computational complexity. We analyzed the computational complexity of two classical decidable problems of algebra

—the uniform word problem for commutative semigroups and the membership problem for polynomials ideals over the rationals. These examples illustrate the significance of questions about the computational complexity of algebraic problems and reveal that methods are available to provide robust answers to such questions. Experience in mathematical logic and automata theory [9, 29] suggests that wherever effective decidability is of interest, analysis of computational complexity can provide further fruitful information. We expect this to be the case also in subsequent studies of algebraic decision problems. We close by listing a few open problems related to the results presented above.

1. What is the computational complexity of *PI*? The reduction of *CSG* to *PI* implies that *PI* is exponential space *hard*, but the best upper bound on the complexity appears to be double exponential or more.

2. Let $\mathscr{P}[\alpha] =_{\text{def}} \{\beta; \alpha \equiv \beta(\mathscr{P})\}$, where $\mathscr{P}$ is a commutative semigroup presentation. Results of [2, 42] imply that it is decidable whether $\mathscr{P}_1[\alpha_1] \subseteq \mathscr{P}_2[\alpha_2]$. What is the complexity of this containment problem?

Vector replacement systems (VRSs), also known as Petri nets of commutative semi-Thue systems, were described in Section 2.

For any VRS $\mathscr{T}$, let $\mathscr{T}[\alpha] =_{\text{def}} \{\beta; \beta \text{ is derivable from } \alpha \text{ in } \mathscr{T}\}$.

3. In [35] it has been shown that the *VRS covering problem*, to decide given $(\alpha, \beta, \mathscr{T})$ whether $\beta\gamma \in \mathscr{T}[\alpha]$ for some word γ, is decidable in space $c^{n^2 \log n}$. Our reduction of *ESC* to *CSG* implies a lower bound of space d^n for some $d > 1$. (This lower bound was originally obtained by Lipton [24].) Improve these bounds.

4. In [28] it has recently been shown that the *VRS reachability problem*, to decide whether $\beta \in \mathscr{T}[\alpha]$, is decidable, but the decision procedure is not primitive recursive [26]. What is the computational complexity of the reachability problem?

5. Another natural problem about finitely presented commutative semigroups is whether two presentations define isomorphic semigroups. This problem is not even known to be decidable [43].

APPENDIX: DEGREE BOUNDS FOR SOLUTIONS OF LINEAR EQUATIONS OVER $\mathbb{Q}[x_1,..., x_v]$

Let

$$\sum_{j=1}^{s} f_{ij} g_j = b_i, \qquad i = 1,..., t, \tag{1}$$

be a system of linear equations with $b_i, f_{ij} \in \mathbb{Q}[x_1,..., x_v]$,

$$\begin{aligned} q &\stackrel{\text{def}}{=} \text{the maximum degree of the } f_{ij}, \\ B &\stackrel{\text{def}}{=} \text{the maximum degree of the } b_i. \end{aligned} \tag{2}$$

Further, let F be the $t \times s$ matrix whose (i, j)th entry is $f_{i,j}$. We assume without loss of generality that the rank of F is $t \leqslant s$ (otherwise, equations can be eliminated from system (1), or (1) has no solution), and that the first t columns of F are linearly independent.

Let $\Delta = \Delta_{1,...,t}$ be the determinant formed from these columns.

By a rational invertible linear transformation of $x_1,..., x_v$ we can transform Δ to be regular in x_1, i.e., the degree $\deg_{x_1}(\Delta)$ of Δ in x_1 equals the degree $\deg(\Delta)$ of Δ. Note that such transformations do not affect the degrees of elements of $\mathbb{Q}[x_1,..., x_v]$. Thus we may assume without loss of generality that Δ is regular in x_1. By Cramer's rule,

$$l_k \stackrel{\text{def}}{=} (\Delta_{t+k,2,...,t}, \Delta_{1,t+k,3,...,t},..., \Delta_{1,2,...,t-1,t+k}, \overbrace{\underbrace{0,..., 0}_{k-1}, -\Delta, 0,..., 0}^{s-t})$$

is, for each $k = 1,..., s-t$, a possible solution of the homogeneous system given by (1). Hence we have for the jth component $(l_k)_j$ of l_k

$$\deg_{x_1}((l_k)_j) \leqslant tq \leqslant sq \qquad \text{for} \quad k = 1,..., s-t, \quad j = 1,..., s. \tag{3}$$

Furthermore, as Δ is regular in x_1, polynomials $c_i \in \mathbb{Q}[x_1,..., x_v]$ can be chosen for $i = 1,..., t$, such that

$$\begin{aligned} \deg(c_i) &\leqslant \deg(b_i) - \deg(\Delta), \\ \deg(b_i - \Delta c_i) &\leqslant \deg(b_i), \\ \deg_{x_1}(b_i - \Delta c_i) &< \deg(\Delta) \leqslant sq. \end{aligned} \tag{4}$$

Consider the system

$$\sum_{j=1}^{s} f_{ij} g'_j = b_i - \Delta c_i, \qquad i = 1,..., t, \tag{5}$$

and let $(g'_1,..., g'_s)$ be a solution of (5). By subtracting appropriate multiples of l_k, for $k = 1,..., s-t$, we can, because of (3), obtain a solution $(g''_1,..., g''_s) \in \mathbb{Q}[x_1,..., x_v]$ with

$$\deg_{x_1}(g''_j) < \deg_{x_1}(\Delta) \leqslant sq \qquad \text{for} \quad j = t+1,..., s. \tag{6}$$

For $b_i' =_{\text{def}} \sum_{j=1}^{t} f_{ij} g_j''$ we have

$$b_i' = b_i - \Delta c_i - \sum_{j=t+1}^{s} f_{ij} g_j''.$$

Now $\deg_{x_1}(b_i - \Delta c_i) < \deg_{x_1}(\Delta)$ by (4) and $\deg_{x_1}(\sum_{j=t+1}^{s} f_{ij} g_j'') < \deg_{x_1}(\Delta) + q$ by (2) and (6). Hence we conclude, again by Cramer's rule, that $\deg_{x_1}(\Delta g_j'') = \deg_{x_1}(\sum_{i=1}^{t} b_i' F_{ij}) < \deg(\Delta) + q + (s-1)q$, where F_{ij} is the (i, j)th minor of F. Hence, for $j = 1,..., s$, $\deg_{x_1}(g_j'') < sq$.

For $g_j =_{\text{def}} g_j'' + \sum_{i=1}^{t} c_i F_{ij}$, $j = 1,..., t$, and $g_j =_{\text{def}} g_j''$ for $j = t+1,..., s$, we therefore obtain a solution $(g_1,..., g_s)$ of (1) with $\deg_{x_1}(g_j) < sq + B$.

Adding the coefficients of all equal powers of x_1 (from x_1^0 to x_1^{q+sq-1})[3] we obtain from (5) at most $t(q + sq)$ equations with $\leqslant s^2 q$ unknowns in $\mathbb{Q}[x_2,..., x_v]$ where the degrees of the coefficients are still bounded by q, and the degrees of the righthand sides by B. For the function $m(s, v, q, B)$ which bounds the minimal degree of solution for (1) we, therefore, get the following recurrence relation:

$$m(s, v, q, B) \leqslant \max\{sq + m(s^2 q, v-1, q, B),\ sq + B\},$$

and hence

$$m(s, v, q, B) \leqslant sq + s^2 q^2 + s^4 q^4 + \cdots + (sq)^{2^{v-1}} + B \leqslant B + (sq)^{2^v}. \quad \blacksquare$$

We observe that the constructions of Section 6 imply that a degree bound growing double exponentially in the number of variables is unavoidable.

References

1. A. V. Aho, J. E. Hopcroft, and J. D. Ullman, "The Design and Analysis of Computer Algorithms," Addison–Wesley, Reading, Mass., 1974.
2. A. P. Biryukov, Some algorithmic problems for finitely defined commutative semigroups, *Siberian Math. J.* **8** (1967), 384–391.
3. M. Blum, A machine-independent theory of the complexity of recursive functions, *J. Assoc. Comput. Mach.* **14** (1967), 322–336.
4. M. Blum, On effective procedures for speeding up algorithms, *J. Assoc. Comput. Mach.* **18** (1971), 290–305.
5. E. W. Cardoza, "Computational Complexity of the Word Problem for Commutative Semigroups," M.I.T., Project MAC, TM 67, October 1975.
6. E. Cardoza, R. Lipton, and A. R. Meyer, Exponential space complete problems for Petri nets and commutative semigroups: Preliminary report, *in* "Proceedings, 8th Ann. ACM STOC (1976)," pp. 50–54.
7. S. A. Cook, The complexity of theorem proving procedures, *in* "Proceedings, 3rd Ann. ACM STOC (1971)," pp. 151–158.

[3] Not x_1^{sq-1} as in [15].

8. V. A. EMILIČEV, On algorithmic decidability of certain mass problems in the theory of commutative semigroups, *Sibirsk. Mat. Ž* **4** (1963), 788–798. [In Russian.]
9. J. FERRANTE AND C. RACKOFF, "The Computational Complexity of Logical Theories," Lecture Notes in Mathematics No. 718, Springer–Verlag, Berlin/Heidelberg/New York, 1979.
10. P. C. FISCHER, A. R. MEYER, AND A. L. ROSENBERG, Counter machines and counter languages, *Math. Systems Theory* **2**(3) (1968), 265–283.
11. S. GINSBURG AND E. H. SPANIER, Semigroups, Presburger formulas, and languages, *Pacific J. Math.* **16**(2) (1965), 285–296.
12. M. HACK, "Petri Nets and Commutative Semigroups," M.I.T., Project MAC, CSGN 18, July 1974.
13. M. HACK, "Decidability Questions for Petri Nets," M.I.T., Laboratory for Computer Science, TR 161, June 1976.
14. K. HENTZELT, Zur Theories der Polynomideale und Resultanten, *Math. Ann.* **88** (1922), 53–79.
15. G. HERMANN, Die Frage der endlich vielen Schritte in der Theorie der Polynomideale, *Math. Ann.* **95** (1926), 736–788.
16. D. HILBERT, Über die Theorie der algebrischen Formen, *Math. Ann.* **36** (1890), 473–534.
17. A. W. HOLT *et al.*, "Final Report of the Information Systems Theory Project," Griffiss Air Force Base, Rome Air Development Center (N.Y.), RADC-TR-68-305 (1968).
18. J. E. HOPCROFT AND J. D. ULLMAN, "Introduction to Automata Theory, Languages, and Computation," Addison–Wesley, Reading, Mass., 1979.
19. J. HARTMANIS AND R. E. STEARNS, On the computational complexity of algorithms, *Trans. Amer. Math. Soc.* **117** (1965), 285–306.
20. R. M. KARP, "Reducibility among Combinatorial Problems. Complexity of Computer Computations," pp. 85–104, Plenum, New York, 1972.
21. R. M. KELLER, "Vector Replacement Systems: A Formalism for Modelling Asynchronous Systems," Princeton University, CSL, TR 117, (December 1972).
22. J. KÖNIG, "Einleitung in die allgemeine Theorie der algebraischen Grössen," Teubner, Leipzig, 1903.
23. R. LAING, "Realization and Complexity of Commutative Events," University of Michigan, TR 03105-48-T (1967).
24. R. LIPTON, "The Reachability Problem is Exponential-Space-Hard," Yale University, Department of Computer Science, Report No. 62, January 1976.
25. A. I. MALCEV, On homomorphisms of finite groups, *Ivanov. Gos. Ped. Inst. Učen. Zap.* **18** (1958), 49–60. [In Russian.]
26. E. W. MAYR AND A. R. MEYER, The Complexity of the finite containment problem for Petri Nets, *J. Assoc. Comput. Mach.* **28**(2) (1981).
27. A. MARKOV, The impossibility of certain algorithms in the theory of associative systems, *Dokl. Akad. Nauk SSSR* **5** (1947), 587–590.
28. E. W. MAYR, An algorithm for the general Petri net reachability problem, *in* "Proceedings, 13th Ann. ACM STOC (1981)," pp. 238–246.
29. A. R. MEYER, The inherent computational complexity of theories of ordered sets, *in* "Proceedings of the International Congress of Mathematicians, 1974," Vol. 2 (1975), pp. 477–482.
30. M. MINSKY, Recursive unsolvability of Post's problem of tag and other topics in the theory of Turing machines. *Ann. Math.* **74** (1961), 437–455.
31. A. R MEYER AND L. STOCKMEYER, The equivalence problem for regular expressions with squaring requires exponential space, *in* "Proceedings, 13th Ann. IEEE Symp. on SWAT (1973)," pp. 125–129.
32. M. MACHTEY AND P. R. YOUNG, "An Introduction to the General Theory of Algorithms," North-Holland, New York, 1978.

33. C. A. PETRI, "Kommunikation mit Automaten," Schriften des IMM No. 2, Institut für Instrumentelle Mathematik (Bonn), 1962.
34. E. POST, Recursive unsolvability of a problem of Thue, *J. Symbolic Logic* **12** (1947), 1–11.
35. C. RACKOFF, The covering and boundedness problems for vector additions systems, *Theoret. Comput. Sci.* **6** (1978), 223–231.
36. F. RICHMAN, Constructive aspects of Noetherian rings, *Proc. Amer. Math. Soc.* **44** (1974), 436–441.
37. W. SAVITCH, Relationships between nondeterministic and deterministic tape complexities, *J. Comput. System Sci.* **4** (1970), 177–192.
38. A. SEIDENBERG, Constructions in algebra, *Trans. Amer. Math. Soc.* **197** (1974), 273–313.
39. H. SIMMONS, The word and torsion problems for commutative Thue systems, *in* "Word Problems II" (S. I. Adian, W. W. Boone, and G. Higman, Eds.), pp. 395–400, North-Holland, New York, 1980.
40. L. STOCKMEYER AND A. R. MEYER, Word problems requiring exponential space, *in* "Proceedings, 5th Ann. ACM STOC (1973)," pp. 1–9.
41. L. STOCKMEYER, "The Complexity of Decision Problems in Automata Theory and Logic," M.I.T., Project MAC, TR 133 (1974).
42. M. A. TAICLIN, Algorithmic problems for commutative semigroups, *Soviet Math. Dokl.* **9** (1968), 201–204.
43. M. A. TAICLIN, On the isomorphism problem for commutative semigroups, *Math. USSR Sb.* **22** (1974), 104–128.