

ANDREAS BRANDSTÄDT

**Space classes, intersection of languages and
bounded erasing homomorphisms**

RAIRO. Informatique théorique, tome 17, n° 2 (1983), p. 121-130

http://www.numdam.org/item?id=ITA_1983__17_2_121_0

© AFCET, 1983, tous droits réservés.

L'accès aux archives de la revue « RAIRO. Informatique théorique » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques

<http://www.numdam.org/>

SPACE CLASSES, INTERSECTION OF LANGUAGES AND BOUNDED ERASING HOMOMORPHISMS (*)

by Andreas BRANDSTÄDT ⁽¹⁾

Communicated by R. V. BOOK

Abstract. — We characterize space complexity classes in terms of bounded erasing homomorphisms applied to threefold intersections of one-counter languages, and also we characterize the classes which are given by bounded erasing homomorphisms applied to mixed intersections of one-counter languages and context-free languages or one-counter languages and checking stack languages. A generalization to erasing bounded transducers is made.

Résumé. — On caractérise les classes de complexité spatiale en termes d'homomorphismes à effacement borné appliqués à des intersections de trois langages à un compteur. On caractérise également les classes obtenues en appliquant des homomorphismes à effacement borné à des intersections de langages à un compteur avec des langages algébriques (context-free) ou à des intersections de langages à un compteur avec des langages vérifiables (checking stack). On donne aussi une généralisation aux transducteurs à effacement borné.

1. INTRODUCTION

There are a lot of results concerning the representation of complexity classes by means of bounded erasing homomorphisms applied to simple families of languages. Consider e. g. the class $\text{NTIME}(n)$ of quasi-realtime languages which is the closure of the intersection of three context-free languages under length-preserving homomorphisms (see Book and Greibach [6] and a refinement in Book/Nivat/Paterson [8]) and the correspondence between the amount of erasing of homomorphisms and time bounds (Book/Greibach/Wegbreit [7]).

More recent results provide a representation of time and space complexity classes by means of bounded erasing homomorphisms applied to equality sets and equality sets with bounded balance (cf. [5, 10]).

The aims of this paper are twofold.

(*) Received in February 1981, revised in December 1981.

(¹) Sektion Mathematik der Friedrich-Schiller-Universität Jena Universitätshochhaus, 6900 Jena, R.D.A.

First we characterize space complexity classes 1-NSPACE(s) of on-line Turing machines by means of bounded erasing homomorphisms and intersections of one-counter languages and characterize the result of applying bounded erasing homomorphisms to mixed intersections of one-counter languages and context-free languages or one-counter languages and checking stack languages, and then we show that the amount of erasing of pushdown and counter machine transductions correspond to the amount of erasing of homomorphisms which are used together with the characterizing languages of transductions.

2. NOTATIONS AND BASIC RESULTS

We assume familiarity with basic concepts from AFL theory and from complexity theory.

Let \mathbb{N} denote the set of positive integers.

Let id denote the function $\text{id}(n) = n$ for all $n \in \mathbb{N}$.

Σ^* denotes the set of all finite strings over an alphabet Σ . A homomorphism $h : \Sigma^* \rightarrow \Delta^*$ is f -bounded erasing on a language $L \subseteq \Sigma^*$ iff there is a constant 0 such that for all $w \in L$ $f(|h(w)|) \geq c \cdot |w|$ where $|w|$ denotes the length of w . For a class of functions \mathcal{F} a homomorphism h is called \mathcal{F} -erasing on L iff there is a function $f \in \mathcal{F}$ such that h is f -bounded erasing on L .

Important classes \mathcal{F} are the polynomials $\text{POL} = \{n^k : k \in \mathbb{N}\}$ and the exponential functions $\text{EXP} = \{2^{c \cdot n} : C > 0 \text{ an arbitrary constant}\}$.

In the same way the classes:

$\text{POL } f = \{f(n)^k : k \in \mathbb{N}\}$ and

$\text{EXP } f = \{2^{c \cdot f(n)} : c > 0 \text{ an arbitrary constant}\}$,

are defined for a function f .

For a family \mathcal{L} of languages let $H_f(\mathcal{L})$ denote the closure of \mathcal{L} under f -bounded erasing homomorphisms and let $H_{\mathcal{F}}(\mathcal{L})$ denote the closure of \mathcal{L} under \mathcal{F} -erasing bounded homomorphisms, i. e.:

$H_f(\mathcal{L}) = \{h(L) : L \in \mathcal{L} \text{ and } h \text{ is a homomorphism which}$
is f -bounded erasing on $L\}$,

and:

$$H_{\mathcal{F}}(\mathcal{L}) = \bigcup_{f \in \mathcal{F}} H_f(\mathcal{L}).$$

A homomorphism $h : \Sigma^* \rightarrow \Delta^*$ is length-preserving iff:

$$|h(w)| = |w| \quad \text{for all } w \in \Sigma^*.$$

Let $H(\mathcal{L}) = \{h(L) : L \in \mathcal{L} \text{ and } h \text{ is a length-preserving homomorphism}\}$ denote the closure of a class \mathcal{L} under length-preserving homomorphisms. Let \hat{H} denote the closure under arbitrary homomorphisms. Let LIN, 1-C, CF, CSA, RE denote the families of linear, one-counter, context-free, (one-way) checking stack, contextsensitive and recursively enumerable languages respectively.

Let \wedge denote the wedge operator from AFL theory, i. e., for families of languages \mathcal{L}_1 and \mathcal{L}_2 :

$$\mathcal{L}_1 \wedge \mathcal{L}_2 = \{L_1 \cap L_2 : L_1 \in \mathcal{L}_1, L_2 \in \mathcal{L}_2\}.$$

Furthermore we define time and space complexity classes for several types of machines (cf. [7]). A machine M of a given type operates within time (space) f iff for each input string w and each accepting computation of M on w the machine M operates within $f(|w|)$ -bounded time (space).

Some particular complexity classes used in this paper are the:

– space complexity classes of on-line deterministic (nondeterministic) Turing machines:

1- T -XSPACE(f) where $f(n) \geq \log n$, and $X \in \{D, N\}$;

– space complexity classes of Turing machines with one tape (and without a one-way input tape):

XSPACE(f) where $f(n) \geq n$, and $X \in \{D, N\}$.

(Obviously for $f \geq \text{id}$ 1- T -XSPACE(f) = XSPACE(f));

– time complexity classes of on-line k -tape nondeterministic Turing machines:

kT -NTIME(f) for $f(n) \geq n$,

$$\text{NTIME}(f) \stackrel{\text{df}}{=} \bigcup_{k=1}^{\infty} kT\text{-NTIME}(f)$$

– time complexity classes of (non) deterministic k -counter machines with a one-way input tape:

1- kC -XTIME(f) for $f(n) \geq n$ and

$$1\text{-multi}C\text{-XTIME}(f) = \bigcup_{k=1}^{\infty} 1\text{-}kC\text{-XTIME}(f), X \in \{D, N\};$$

– time-space complexity classes of on-line Turing machines with an additional unbounded pushdown tape (the tape is in fact only bounded by the time-bound):

1-aux PD - T -NSPACE-TIME(f, g) (cf. [17]);

– time-space complexity classes of on-line Turing machines with an

additional unbounded checking stack tape:

1-aux CSA- T -NSPACE-TIME(f, g) (cf. [16]).

We recall some theorems from [6, 8 and 7].

PROPOSITION:

NTIME(n) = $H(\text{CF} \wedge \text{CF} \wedge \text{CF})$ [6]

= $H(\text{CF} \wedge \text{LIN} \wedge \text{LIN})$ [8].

NTIME(f) = $H_f(\text{NTIME}(n))$ for $f \geq \text{id}$,

NSPACE(f) = $H_f(\text{NSPACE}(n))$ for $f \geq \text{id}$,

and DSPACE(f) = $H_f(\text{DSPACE}(n))$ for $f \geq \text{id}$. [7].

3. THE REPRESENTATION OF NONDETERMINISTIC SPACE COMPLEXITY CLASSES

Our representation is based on three lemmas.

LEMMA 1 [11]:

1- T -XSPACE($\log t$) = 1-3C-XTIME(POL t), $X \in \{D, N\}$, $t(n) \geq n$.

LEMMA 2 [7]:

1- k C-NTIME(Pol t) = $H_{\text{Pol } t}(1-k\text{C-NTIME}(n))$, $t(n) \geq n$.

The corresponding assertion in [7] is formulated for Turing machines but the padding method used for the proof in [7] can be applied easily to time-bounded counter machines and to many other types of multitape machines.

LEMMA 3:

1- k C-NTIME(n) = $H(\underbrace{1-C \wedge \dots \wedge 1-C}_{k \text{ times}})$.

Lemma 3 is a consequence of the corresponding representation of quasi-realtime multi-tape AFA (see [12]) and the fact that 1-C equals the class of languages acceptable in real-time by nondeterministic one-counter machines [13]. It is easy to see that 1-C can be replaced by the deterministic class 1-DC thus having 1- k C-NTIME(n) = $H(\underbrace{1-DC \wedge \dots \wedge 1-DC}_{k \text{ times}})$.

This is possible also for Q and other classes, e. g.:

$$Q = H(\text{DCF} \wedge \text{DCF} \wedge \text{DCF}).$$

From lemma 1, 2 and 3 we obtain our first representation theorem.

THEOREM 1:

$1-T-NSPACE(s) = H_{EXP_s}(1-DC \wedge 1-DC \wedge 1-DC)$ for $s \geq \log$.

(This means $NSPACE(s) = H_{EXP_s}(1-DC \wedge 1-DC \wedge 1-DC)$ for $s \geq \text{id}$.)

Since $NTIME(n) = H(DCF \wedge DCF \wedge DCF)$ [6] and

$NTIME(t) = H_t(NTIME(n))$ [7] it follows that

$NTIME(t) = H_t(DCF \wedge DCF \wedge DCF)$.

This representation can be improved if a class \mathcal{F} of bounding functions is closed under squaring, i. e. if $f \in \mathcal{F}$ then $(f(n))^2 \in \mathcal{F}$ (e. g. the classes POL or EXP).

THEOREM 2: Let \mathcal{F} be a class of functions which is closed under squaring.

Then $NTIME(\mathcal{F}) = H_{\mathcal{F}}(DCF \wedge DCF)$.

The proof bases on a representation of Turing machine computations given by Hartmanis in [14], where $\hat{H}(CF \wedge CF) = RE$ is shown. This relation can be extended to complexity restrictions. In [2] the representation of [14] is refined to $RE = \hat{H}(\text{LIN} \wedge \text{LIN})$. We can extend this representation to complexity bounds obtaining

COROLLARY 1: Assume that \mathcal{F} is closed under squaring.

Then $1-kT-NTIME(\mathcal{F}) = H_{\mathcal{F}}(\text{LIN} \wedge \text{LIN})$ holds for classes of bounding functions \mathcal{F} with $f(n) \geq n$ and $(f(n))^2 \in \mathcal{F}$ for all $f \in \mathcal{F}$.

Theorem 2 shows that it is sufficient to have one intersection if the class of bounding functions is closed under squaring.

Is this true also for space complexity classes?

We only know

THEOREM 3: For every bounding function s with $s(n) \geq \log n$.

$1-T-NSPACE(s) \subseteq H_{EXP_{EXP_s}}(1-DC \wedge 1-DC)$ holds.

Proof: Lemma 1 bases on an encoding of a tape bounded by s by three counters whose length is bounded by EXP_s . The contents i, j, k of these three counters can be encoded as $2^i 3^j 5^k$ using two counters. Obviously the length of the two counters is then bounded by EXP_{EXP_s} and also the time necessary for these simulations is bounded by EXP_{EXP_s} .

Now lemma 2 and 3 are applicable and theorem 3 is proved. It is an open question whether this amount of erasing is necessary.

The time-space problem.

$NSPACE(s) \stackrel{?}{\subseteq} NTIME(EXP_s)$ can be reformulated as

$$H_{EXP_s}(1-DC \wedge 1-DC \wedge 1-DC) \stackrel{?}{\subseteq} H_{EXP_s}(DCF \wedge DCF \wedge DCF) \\ (= H_{EXP_s}(DCF \wedge DCF)).$$

This leads to the main results of this paper:

1. We investigate which class is obtained if we substitute one of the components 1-DC by DCF, i. e., what is $H_{EXP_s}(1-DC \wedge 1-DC \wedge DCF)$.

2. We give a simple class of languages (namely real-time checking stack languages) which can be substituted for one component 1-DC to get $H_{EXP_s}(DCF \wedge DCF)$.

THEOREM 4:

1-aux PD-*T*-NSPACE-TIME(s, EXP_s) = $H_{EXP_s}(DCF \wedge 1-DC \wedge 1-DC)$.

Proof: The proof bases on the padding argument described in [7] which was used in lemma 2.

1. " \supseteq ": Assume that $L \in H_{EXP_s}(DCF \wedge 1-DC \wedge 1-DC)$.

Then there is a constant c , languages $L_1 \in DCF$, $L_2, L_3 \in 1-DC$ and a homomorphism h which is $2^{c \cdot s^{(n)}}$ -bounded erasing on $L_1 \cap L_2 \cap L_3$ such that $L = h(L_1 \cap L_2 \cap L_3)$, i. e., $w \in L$ iff there is a $w' \in L_1 \cap L_2 \cap L_3$ with $h(w') = w$. Since h is $2^{c \cdot s^{(n)}}$ -bounded erasing there is a constant c' such that $c' \cdot |w| \leq 2^{c \cdot s^{(|w|)}}$. Now a Turing machine M with one-way input tape, a Turing tape and a pushdown tape accepting L works as follows:

M guesses an input w' symbol by symbol and tests simultaneously $w' \in L_1$ by its pushdown tape, $w' \in L_2 \cap L_3$ by its Turing tape in space s and tests $h(w') = w$. It is straightforward that this can be done in time bounded by $2^{c'' \cdot s}$ for an appropriate constant c'' .

2. " \subseteq ": Let $L \in 1\text{-aux PD-}T\text{-NSPACE-TIME}(s, EXP_s)$. There is a machine M with a s -bounded Turing tape and an auxiliary pushdown tape which accepts L operating in time EXP_s . We want to simulate the work of M by a machine M' with two counters and a pushdown tape operating in time EXP_s . Because the Turing tape of M is s -bounded it can be simulated by three counters in time EXP_s (cf. lemma 1). After simulating one step of the pushdown store of M by the pushdown store of M' the machine M' sets a marker on its pushdown store and uses the pushdown as an additional counter thus having three counters for the simulation of one step of the Turing tape of M . Because after some transformations the encoded content of the Turing tape of M is stored only in two counters the pushdown store of M' is free again to simulate one step of the pushdown of M and so on. Obviously the whole simulation can be done in time EXP_s . Now a slight generalization of lemma 2 and 3 to the case of time-bounded machines with one pushdown store and two counters gives the assertion of theorem 4.

Of course theorem 4 gives no answer to the question $H_{EXP_s}(1-DC \wedge 1-DC \wedge DCF) = H_{EXP_s}(DCF \wedge DCF)$ but if we use a checking stack tape instead of a pushdown tape we have equality:

THEOREM 5:

$$\text{NTIME}(\text{EXP}_s) = H_{\text{EXP}_s}(\text{DCF} \wedge \text{DCF}) = H_{\text{EXP}_s}(\text{1-DC} \wedge \text{DCSA}).$$

(1)

Proof: The inclusion " \supseteq " in (1) is obvious. For the other inclusion " \subseteq " we remark that the checking stack which works within time EXP_s guesses the accepting computation sequence of configurations of the corresponding Turing machine which works in time EXP_s and then by 1 counter it is tested whether this sequence describes an accepting computation of a Turing machine. Obviously all this can be done in time EXP_s .

Theorem 5 is closely related to a result on space complexity classes with and without an auxiliary checking stack tape shown by Ibarra [16].

4. BOUNDED ERASING TRANSDUCTIONS

In the previous section bounded erasing homomorphisms have been applied to certain classes of languages. Pushdown (and other) transductions are a well-known concept of formal language theory (cf. [1]). They are characterized by so-called "characterizing languages" (in the case of pushdown transductions these are context-free languages) and pairs of homomorphisms applied to the characterizing languages.

We investigate here the result of applying pairs of bounded erasing homomorphisms in the representation of transductions. This is the content of sections 4 and 5. Hereby it turns out that the padding method known from [7] is also applicable in these cases. We describe the notions used here only informally. A pushdown transducer is a pushdown automaton with an (one-way) output tape (see [1]).

The transduction $\tau(P)$ computed by a pushdown transducer P is the set of pairs (x, y) , for which there is a computation of P where P starting on the input word x in the initial state and with empty pushdown and output tape comes to a final state when the input x is read, and the output tape contains y . This is the notion used in [1].

In the case where the pushdown store has only one symbol the pushdown transducer is a counter transducer.

A pushdown transducer P is f -bounded erasing on L iff there is a constant c such that for all $(u, v) \in \tau(P)$ with $u \in L$, $c \cdot f(|v|) \geq |u|$ holds.

Correspondingly f -bounded erasing transductions are defined. Furthermore we use the notion of characterizing languages from [1]: Let ε denote the empty string.

(a) A language $L \subseteq (\Sigma \cup \Delta)^*$ characterizes the transduction τ , $\tau \subseteq \Sigma^* \times \Delta^*$ iff there are homomorphisms h_1, h_2 such that $\tau = \{(h_1(w), h_2(w)) : w \in L\}$.

Let $\Delta' = \{b' : b \in \Delta\}$ and suppose that $\Delta \cap \Delta' = \emptyset$.

(b) A language $L \subseteq (\Sigma \cup \Delta)^*$ strongly characterizes the transduction τ , $\tau \subseteq \Sigma^* \times \Delta^*$ iff:

$$1) \Sigma \cap \Delta' = \emptyset,$$

$$2) \tau = \{(h_1(w), h_2(w)) : w \in L\}, \text{ where:}$$

$$(\alpha) h_1(a) = a \text{ for all } a \in \Sigma, h_1(b) = \varepsilon \text{ for all } b \in \Delta';$$

$$(\beta) h_2(a) = \varepsilon \text{ for all } a \in \Sigma, h_2(b') = b \text{ for all } b' \in \Delta'.$$

The following property can be found in [1].

PROPOSITION: τ is a pushdown transduction iff τ is strongly characterizable by a context-free language.

This property was generalized by Ibarra [15] to the case of AFA and AFT and corresponding characterization languages given by the underlying AFA.

Now we generalize this proposition to the case of bounded erasing.

THEOREM 6: The following properties are equivalent:

1) The pushdown (one-counter) transduction τ is f -bounded erasing on \hat{L} .

2) There is a context-free (one-counter) language which characterizes τ and for the corresponding homomorphisms $h_1, h_2, c.f(|h_2(w)|) \geq |h_1(w)|$ holds for an appropriate constant c .

3) There is a context-free (one-counter) language L which strongly characterizes τ and for the corresponding homomorphisms h_1, h_2 there is an appropriate constant c such that $c.f(|h_2(w)|) \geq |w|$ holds for all $w \in L \cap h_1^{-1}(\hat{L})$.

The proof of this theorem is a straightforward application of the padding method described in [7].

5. TIME-BOUNDED TRANSDUCTIONS

In this section we investigate Turing machine transducers, which have a one-way input tape, a Turing tape as storage tape and a one-way output tape. The concept is the same as in section 4 with Turing machines instead of pushdown automata.

DEFINITION: Let τ be a Turing machine transduction and M be a Turing machine transducer (with one working tape) which computes τ . τ is nondeterministically computable within time bounded by t iff for all $(u, v) \in \tau$

there is a computation path of M where M starting on u processes the output v and enters a final state (an accepting path) and every accepting path has a length of no more than $t(|u|)$ steps.

THEOREM 7: *The following properties are equivalent:*

- 1) τ is a Turing machine transduction which is time-bounded by t .
- 2) There is a characterizing language $L \in \text{NTIME}(n)$ and homomorphisms h_1, h_2 such that:

$$\tau = \{ (h_1(w), h_2(w)) : w \in L \text{ and } h_1 \text{ is } f\text{-bounded erasing on } L \}.$$

- 3) There is a language $L \in \text{NTIME}(f)$ and homomorphisms h_1, h_2 such that τ is strongly characterized by L and h_1, h_2 .

The proof of theorem 7 is also a straightforward application of the padding method from [7].

ACKNOWLEDGMENT

I wish to express my gratitude to Dr. Franz J. Brandenburg and Dr. Klaus Wagner for several hints and helpful comments.

REFERENCES

1. A. V. AHO and J. D. ULLMAN, *The Theory of Parsing, Translation and Compiling*, Vol. 1, Prentice-Hall, Englewood Cliffs N.J., 1972.
2. B. S. BAKER and R. V. BOOK, *Reversal-Bounded Multipushdown Machines*, J. Comp. System Sc., Vol. 8, 1974, pp. 315-332.
3. R. V. BOOK, *Translational Lemmas, Polynomial Time and $(\log n)^j$ -Space*, Theor. Comp. Sc., 1976, pp. 215-226.
4. R. V. BOOK, *Complexity Classes of Formal Languages*, MFCS, 1979, pp. 43-56, Lecture Notes in Comp. Sc., No. 74.
5. R. V. BOOK and F.-J. BRANDENBURG, *Equality Sets and Complexity Classes*, S.I.A.M. J. Computing, Vol. 9, No. 4, 1980, pp. 729-743.
6. R. V. BOOK and S. A. GREIBACH, *Quasi-Realtime Languages*, Math. Syst. Theory, Vol. 4, No. 2, 1970, pp. 97-111.
7. R. V. BOOK, S. A. GREIBACH and B. WEGBREIT, *Time- and Tape-Bounded Turing Acceptors and AFLs*, J. Comp. System Sc., Vol. 4, No. 6, 1970, pp. 606-621.
8. R. V. BOOK, M. NIVAT and M. PATERSON, *Reversal-Bounded Acceptors and Intersections of Linear Languages*, S.I.A.M. J. Computing, Vol. 3, No. 4, 1974, pp. 283-295.
9. K. CULIK II, *Homomorphisms; Decidability, Equality and Test Sets; Formal Language Theory, Perspectives and Open Problems*, R. V. Book, Ed., Academic Press, 1980.
10. K. CULIK II and N. O. DIAMOND, *A Homomorphic Characterization of Time and Space Complexity Classes of Languages*, Internat. J. Computer Math. (to appear).

11. P. C. FISCHER, A. R. MEYER and A. L. ROSENBERG, *Counter Machines and Counter Languages*, Math. Syst. Theory, Vol. 2, No. 3, 1968, pp. 265-283.
12. S. GINSBURG, *Algebraic and Automata-Theoretic Properties of Formal Languages*, North-Holland, 1975.
13. S. A. GREIBACH, *Remarks on the Complexity of Nondeterministic Counter Languages*, Theor. Comp. Sc., Vol. 1, 1976, pp. 269-288.
14. J. HARTMANIS, *Context-Free Languages and Turing Machine Computations*, Proc. Symp. Applied Math., Vol. 19, 1967, pp. 42-51.
15. O. M. IBARRA, *Characterizations of Transductions Defined by Abstract Families of Transducers* Math. Syst. Theory, Vol. 5, 1971, pp. 271-281.
16. O. M. IBARRA, *Characterization of Some Tape and Time Complexity Classes of Turing Machines in Terms of Multi-Head and Auxiliary Stack Automata*, J. Comput. System. Sc., Vol. 5, 1971, pp. 88-117.
17. I. H. SUDBOROUGH, *On the Tape Complexity of Deterministic Context-Free Languages*, J. Assoc. Comp. Mach., Vol. 25, 1978, pp. 405-414.