Bimachines: an introduction

Pedro V. Silva

University of Porto

June 12, 2006

This is part of a joint work with John Rhodes summarized in the preprint *An algebraic analysis of Turing machines and Cook's Theorem leading to a profinite fractal differential equation and a random walk on a deterministic Turing machine.*

The authors thank Jean-Camille Birget for his advice and helpful comments.

lp-mappings

A_1, A_2 - finite nonempty alphabets

A_1^+ - free semigroup on A_1

$\alpha : A_1^+ \to A_2^+$ is an *lp-mapping* if

$$|\alpha(w)| = |w|$$

for every $w \in A_1^+$.

Factorizing the output

For $i = 1, \ldots, |w|$, write

$$w = \lambda_i(w) \cdot \sigma_i(w) \cdot \mu_i(w)$$

with $|\lambda_i(w)| = i - 1$, $\sigma_i(w) \in A_1$.

Example:

w						
a	b	c	d	a	b	c

a b c d	a	b c
$\lambda_5(w)$	$\sigma_5(w)$	$\mu_5(w)$

A domain extension

The lp-mapping $\alpha : A_1^+ \to A_2^+$ induces a mapping

$$\begin{array}{rcl} A_1^* \times A_1 \times A_1^* & \to & A_2 \\ (u, a, v) & \mapsto & \sigma_{|u|+1}\alpha(uav) \end{array}$$

u	a	v

$\alpha \downarrow$

...	b	...

$\alpha : A^+ \to A'^+$ is uniquely determined by
$\alpha(\_, \_, \_) : A^* \times A \times A^* \to A'$ and vice-versa.

Right automata

A *right A-automaton* is a triple $\mathcal{A}_R = (I_R, Q_R, S_R)$ where

- Q_R is a set
- $I_R \in Q_R$
- S_R is an A-semigroup acting on Q_R on the right,so

$$(q_R s_R)s'_R = q_R(s_R s'_R).$$

The action is *proper* ($I_R \notin I_R S_R$) but not necessarily *faithful* (different elements of S_R may have the same action).

The action of S_R on Q_R induces an obvious action of A^* on Q_R.

Morphisms of right automata

Let

$$\mathcal{A}_R = (I_R, Q_R, S_R), \quad \mathcal{A}'_R = (I'_R, Q'_R, S'_R)$$

be right A-automata. A morphism $\varphi : \mathcal{A}_R \to \mathcal{A}'_R$ is defined, whenever S'_R is a quotient of S_R, via a mapping $\varphi : Q_R \to Q'_R$ such that

- $\varphi(I_R) = I'_R$;
- $\varphi(q_R u) = \varphi(q_R)u$ for all $q_R \in Q_R$ and $u \in A^+$.

This is equivalent to say that there exists a mapping on the states and an A-semigroup morphism preserving initial state and the action.

Left automata

A *left A-automaton* is a triple $\mathcal{A}_L = (S_L, Q_L, I_L)$ where

- Q_L is a set
- $I_L \in Q_L$
- S_L is an A-semigroup acting on Q_L on the left,so

$$s_L(s'_L q_L) = (s_L s'_L) q_L.$$

The action is proper ($I_L \notin S_L I_L$) but not necessarily faithful.

The action of S_L on Q_L induces an obvious action of A^* on Q_L.

Bimachines

An A_1, A_2-*bimachine* is a structure of the form

$$\mathcal{B} = ((I_R, Q_R, S_R), f, (S_L, Q_L, I_L)),$$

where

- (I_R, Q_R, S_R) is a right A-automaton;
- (S_L, Q_L, I_L) is a left A-automaton;
- $f : Q_R \times A \times Q_L \to A'$ a full map (the *output function*).

We say that $\mathcal{B}$ is finite if both state sets and semigroups are finite.

Morphisms of bimachines

Let

$$\mathcal{B} = ((I_R, Q_R, S_R), f, (S_L, Q_L, I_L))$$

$$\mathcal{B}' = ((I'_R, Q'_R, S'_R), f', (S'_L, Q'_L, I'_L))$$

be A_1, A_2-bimachines. We say that $\varphi : \mathcal{B} \to \mathcal{B}'$ is a morphism of A_1, A_2-bimachines if $\varphi = (\varphi_R, \varphi_L)$, where

- $\varphi_R : (I_R, Q_R, S_R) \to (I'_R, Q'_R, S'_R)$ is a morphism of right A-automata;
- $\varphi_L : (S_L, Q_L, I_L) \to (S'_L, Q'_L, I'_L)$ is a morphism of left A-automata;
- $\forall u, v \in A^* \; \forall a \in A \quad f'(I'_R u, a, vI'_L) = f(I_R u, a, vI_L)$.

From bimachines to lp-mappings

We associate an lp-mapping

$$\alpha_{\mathcal{B}} : A_1^+ \longrightarrow {A_2}^+$$

to the A_1, A_2-bimachine

$$\mathcal{B} = ((I_R, Q_R, S_R), f, (S_L, Q_L, I_L))$$

by

$$\alpha_{\mathcal{B}}(u, a, v) = f(I_R u, a, vI_L) \quad (u, v \in A^*, a \in A).$$

Thus

$$\alpha_{\mathcal{B}}(w) = \prod_{i=1}^{|w|} f(I_R\lambda_i(w), \sigma_i(w), \mu_i(w)I_L).$$

From lp-mappings to bimachines

Let $\alpha : A_1^+ \to {A_2}^+$ be an lp-mapping.

Proposition. There exists an A_1, A_2-bimachine $\mathcal{B}_\alpha$ such that:

(i) $\alpha_{\mathcal{B}_\alpha} = \alpha$.

(ii) If $\mathcal{B}'$ is a trim A, A'-bimachine such that $\alpha_{\mathcal{B}'} = \alpha$, then there exists a (surjective) morphism $\varphi : \mathcal{B}' \to \mathcal{B}_\alpha$.

(iii) Up to isomorphism, $\mathcal{B}_\alpha$ is the unique trim A, A'-bimachine satisfying (ii).

We can view $\mathcal{B}_\alpha$ as the *minimum* bimachine of α.

The block product

Let

$$\mathcal{B}^{(i)} = ((I_R^{(i)}, Q_R^{(i)}, S_R^{(i)}), f^{(i)}, (S_L^{(i)}, Q_L^{(i)}, I_L^{(i)}))$$

be an A_i, A_{i+1}-bimachine for $i = 1, 2$. We shall define an A_1, A_3-bimachine

$$\mathcal{B}^{(2)} \Box \mathcal{B}^{(1)} = \mathcal{B}^{(21)} = ((I_R^{(21)}, Q_R^{(21)}, S_R^{(21)}), f^{(21)}, (S_L^{(21)}, Q_L^{(21)}, I_L^{(21)}))$$

called the *block product* of $\mathcal{B}^{(2)}$ and $\mathcal{B}^{(1)}$.

The block product turns out to be a construction on bimachines appropriate do deal with composition.

Notation

The block product construction involves sets of mappings whose domain is often a direct product of the form $Q_R^{(1)} \times Q_L^{(1)}$. We shall use the notation $q_R^{(1)} g q_L^{(1)} = g(q_R^{(1)}, q_L^{(1)})$ for $g \in U^{Q_R^{(1)} \times Q_L^{(1)}} = {}^{Q_R^{(1)}}U^{Q_L^{(1)}}$, $q_R^{(1)} \in Q_R^{(1)}$ and $q_L^{(1)} \in Q_L^{(1)}$.

To be consistent, we shall write maps with domains of type $Q_R^{(1)}$ on the right and type $Q_L^{(1)}$ on the left.

The big semigroups

We define

$$\overline{S_R^{(21)}} = \begin{pmatrix} S_L^{(1)} & 0 \\ {}^{Q_R^{(1)}} S_R^{(2)} {}^{Q_L^{(1)}} & S_R^{(1)} \end{pmatrix}.$$

$\overline{S_R^{(21)}}$ is a semigroup for the product

$$\begin{pmatrix} s_L^{(1)} & 0 \\ g & s_R^{(1)} \end{pmatrix} \begin{pmatrix} s'^{(1)}_L & 0 \\ g' & s'^{(1)}_R \end{pmatrix} = \begin{pmatrix} s_L^{(1)} s'^{(1)}_L & 0 \\ g s'^{(1)}_L + s_R^{(1)} g' & s_R^{(1)} s'^{(1)}_R \end{pmatrix},$$

where

$$q_R^{(1)}(g s'^{(1)}_L + s_R^{(1)} g') q_L^{(1)} = (q_R^{(1)} g(s'^{(1)}_L q_L^{(1)})) + ((q_R^{(1)} s_R^{(1)}) g' q_L^{(1)}).$$

The states

Let

$$Q_R^{(21)} = {Q_R^{(2)}}^{Q_L^{(1)}} \times Q_R^{(1)}.$$

It will be often convenient to represent the elements of $Q_R^{(21)}$, termed *R-generalized 2 step crossing sequences*, as 1×2 matrices.

Let

$$I_R^{(21)} = (\gamma_0^{(21)}, I_R^{(1)}),$$

where $\gamma_0^{(21)} \in {Q_R^{(2)}}^{Q_L^{(1)}}$ is defined by $\gamma_0^{(21)}(q_L^{(1)}) = I_R^{(2)}$.

The action

The semigroup $\overline{S_R^{(21)}}$ acts on $Q_R^{(21)}$ on the right by

$$\begin{pmatrix} \gamma & q_R^{(1)} \end{pmatrix} \begin{pmatrix} s_L^{(1)} & 0 \\ g & s_R^{(1)} \end{pmatrix} = \begin{pmatrix} \gamma s_L^{(1)} \cdot q_R^{(1)} g & q_R^{(1)} s_R^{(1)} \end{pmatrix},$$

where

$$(\gamma s_L^{(1)} \cdot q_R^{(1)} g)(q_L^{(1)}) = \gamma(s_L^{(1)} q_L^{(1)}) \cdot q_R^{(1)} g q_L^{(1)}.$$

This is again a form of matrix multiplication (but we refrain from using $+$ for the action).

Cutting to generators

The semigroup $\overline{S_R^{(21)}}$ is not an A_1-semigroup, so let $\eta_R : A^+ \to \overline{S_R^{(21)}}$ be the homomorphism defined by

$$\eta_R(a) = \begin{pmatrix} a_{S_L^{(1)}} & 0 \\ g_a^{(1)} & a_{S_R^{(1)}} \end{pmatrix},$$

where

$$q_R^{(1)} g_a^{(1)} q_L^{(1)} = (f^{(1)}(q_R^{(1)}, a, q_L^{(1)}))_{S_R^{(2)}}$$

for all $q_R^{(1)} \in Q_R^{(1)}$ and $q_L^{(1)} \in Q_L^{(1)}$. We define

$$S_R^{(21)} = \eta_R(A^+).$$

Cutting to generators

For $w \in A^+$, we may write

$$\eta_R(w) = \begin{pmatrix} w_{S_L^{(1)}} & 0 \\ g_w^{(1)} & w_{S_R^{(1)}} \end{pmatrix}$$

for some $g_w^{(1)} \in {}^{Q_R^{(1)}}S_R^{(2)\,Q_L^{(1)}}$.

The duals

Dually, we define

$$Q_L^{(21)} = Q_L^{(1)} \times {}^{Q_R^{(1)}}Q_L^{(2)},$$

$$S_L^{(21)} = \{ \begin{pmatrix} w_{S_L^{(1)}} & 0 \\ & \\ h_w^{(1)} & w_{S_R^{(1)}} \end{pmatrix} \mid w \in A_1^* \}$$

with $h_w^{(1)} \in {}^{Q_R^{(1)}}S_L^{(2)\,Q_L^{(1)}}$. The action is defined by

$$\begin{pmatrix} s_L^{(1)} & 0 \\ h & s_R^{(1)} \end{pmatrix} \begin{pmatrix} q_L^{(1)} \\ \delta \end{pmatrix} = \begin{pmatrix} s_L^{(1)} q_L^{(1)} \\ \\ h q_L^{(1)} \cdot s_R^{(1)} \delta \end{pmatrix}.$$

The output function

The output function $f^{(21)} : Q_R^{(21)} \times A_1 \times Q_L^{(21)} \to A_3$ is defined by

$$f^{(21)}(\begin{pmatrix} \gamma & q_R^{(1)} \end{pmatrix}, a, \begin{pmatrix} q_L^{(1)} \\ \delta \end{pmatrix})$$

$$= f^{(2)}(\gamma(aq_L^{(1)}), f^{(1)}(q_R^{(1)}, a, q_L^{(1)}), (q_R^{(1)}a)\delta).$$

This completes the definition of the bimachine $\mathcal{B}^{(2)} \Box \mathcal{B}^{(1)}$.

If $\mathcal{B}^{(2)}$ and $\mathcal{B}^{(1)}$ are both finite, so is $\mathcal{B}^{(2)} \Box \mathcal{B}^{(1)}$.

The mappings $g_u^{(1)}$ and $h_u^{(1)}$

Let $w \in A^+$, $q_R^{(1)} \in Q_R^{(1)}$ and $q_L^{(1)} \in Q_L^{(1)}$. Write

$$z = \prod_{i=1}^{|w|} f^{(1)}(q_R^{(1)}\lambda_i(w), \sigma_i(w), \mu_i(w)q_L^{(1)}).$$

Then

$$q_R^{(1)} g_w^{(1)} q_L^{(1)} = z_{S_R^{(2)}},$$

$$q_R^{(1)} h_w^{(1)} q_L^{(1)} = z_{S_L^{(2)}}.$$

Composition

Theorem. Let $\mathcal{B}^{(1)}$ be an A_1, A_2-bimachine and let $\mathcal{B}^{(2)}$ be an A_2, A_3-bimachine. Then $\alpha_{\mathcal{B}^{(2)} \square \mathcal{B}^{(1)}} = \alpha_{\mathcal{B}^{(2)}} \alpha_{\mathcal{B}^{(1)}}$.

The block product of faithful bimachines is not necessarily faithful.

Additional properties

Proposition. Let $\mathcal{B}^{(1)}$ be an A_1, A_2-bimachine and let $\mathcal{B}^{(2)}$ and $\mathcal{B}'^{(2)}$ be A_2, A_3-bimachines. Let $\varphi^{(2)} : \mathcal{B}^{(2)} \to \mathcal{B}'^{(2)}$ be a morphism. Then there exists a morphism $\varphi^{(21)} : \mathcal{B}^{(2)} \square \mathcal{B}^{(1)} \to \mathcal{B}'^{(2)} \square \mathcal{B}^{(1)}$ naturally induced by $\varphi^{(2)}$.

Proposition. Let $\mathcal{B}^{(i)}$ be an A_i, A_{i+1}-bimachine for $i = 1, 2$. Then there exist canonical surjective homomorphisms

$$\xi_R^{(21)} : (I_R^{(21)}, Q_R^{(21)}, S_R^{(21)}) \to (I_R^{(1)}, Q_R^{(1)}, S_R^{(1)}),$$

$$\xi_L^{(21)} : (S_L^{(21)}, Q_L^{(21)}, I_L^{(21)}) \to (S_L^{(1)}, Q_L^{(1)}, I_L^{(1)}).$$

And then there were three...

Let

$$\mathcal{B}^{(i)} = ((I_R^{(i)}, Q_R^{(i)}, S_R^{(i)}), f^{(i)}, (S_L^{(i)}, Q_L^{(i)}, I_L^{(i)}))$$

be an A_i, A_{i+1}-bimachine for $i = 1, 2, 3$. Write

$$\mathcal{B}^{(3(21))} = \mathcal{B}^{(3)} \Box (\mathcal{B}^{(2)} \Box \mathcal{B}^{(1)}), \quad \mathcal{B}^{((32)1)} = (\mathcal{B}^{(3)} \Box \mathcal{B}^{(2)}) \Box \mathcal{B}^{(1)}.$$

We can get associativity at the semigroup level (for three bimachines, but not necessarily for four bimachines!).

Lemma. $S_R^{(3(21))} \cong S_R^{((32)1)}$ and $S_L^{(3(21))} \cong S_L^{((32)1)}$.

A cardinality argument

We have

$$|Q_R^{(3(21))}| > |Q_R^{((32)1)}|.$$

Follows from

$$a^{bc^d}e^b d > (ac^e)^b d.$$

Similarly,

$$|Q_L^{(3(21))}| > |Q_L^{((32)1)}|.$$

Let $\varphi_R : Q_R^{(3(21))} \to Q_R^{((32)1)}$ be defined as follows. Given

A morphism

$$(\gamma^{(3(21))},(\gamma^{(21)},q_R^{(1)}))\in {Q_R^{(3)}}^{Q_L^{(21)}}\times({Q_R^{(2)}}^{Q_L^{(1)}}\times Q_R^{(1)})=Q_R^{(3(21))},$$

we set

$$(\gamma^{(3(21))},(\gamma^{(21)},q_R^{(1)}))\varphi_R=(\gamma^{((32)1)},q_R^{(1)})\in {Q_R^{(32)}}^{Q_L^{(1)}}\times Q_R^{(1)}=Q_R^{((32)1)},$$

where

$$\gamma^{((32)1)}(q_L^{(1)})=(\beta_{q_L^{(1)}},\gamma^{(21)}(q_L^{(1)}))\in {Q_R^{(3)}}^{Q_L^{(2)}}\times Q_R^{(2)}=Q_R^{(32)}$$

and

$$\beta_{q_L^{(1)}}(q_L^{(2)})=\gamma^{(3(21))}(q_L^{(1)},\overline{q_L^{(2)}}),$$

Half associativity

where $\overline{q_L^{(2)}} \in {}^{Q_R^{(1)}}Q_L^{(2)}$ is the constant mapping with image $q_L^{(2)}$.

Theorem. $(\mathcal{B}^{(3)}\Box\mathcal{B}^{(2)})\Box\mathcal{B}^{(1)}$ is a quotient of $\mathcal{B}^{(3)}\Box(\mathcal{B}^{(2)}\Box\mathcal{B}^{(1)})$.

We shall choose bracketing from left to right, that is, priority is assumed to hold from left to right:

$$((\ldots(\mathcal{B}^{(n)}\Box\mathcal{B}^{(n-1)})\Box\mathcal{B}^{(n-2)})\Box\ldots)\Box\mathcal{B}^{(1)}.$$

Our model (informal)

We are interested in deterministic Turing machines that halt for all inputs, particularly those that can solve NP-complete problems. In comparison with the most standard models, our model presents three particular features:

- the "tape" is potentially infinite in *both* directions and has a distinguished cell named *the origin*;
- the origin contains the symbol $\#$ until the very last move of the computation, and $\#$ appears in no other cell;
- the machine always halts in one of a very restricted set of configurations.

Our model (formal)

Our deterministic Turing machine is then a quadruple of the form $\mathcal{T} = (Q, q_0, A, \delta)$ where

- Q is a finite set (set of states) containing the initial state q_0;
- A is a finite set (restricted tape alphabet) containing the special symbols B (blank), B' (pseudoblank), Y (yes), N (no), G (garbage) and $\#$ (origin);
- δ is a union of full maps

$$Q \times (A \setminus \{\#\}) \to Q \times (A \setminus \{\#, B, Y, N, G\}) \times \{L, R\},$$

$$Q \times \{\#\} \to (Q \times \{\#\} \times \{L, R\}) \cup \{Y, N, G\}.$$

Symbols

We write $A^o = A \setminus \{\#, B\}$.

Since the machine is not allowed to write blanks (we must consider space functions), we shall use the pseudoblank as a substitute to keep the final configurations simple.

Note that, in the final move of a computation, the control head is removed from the tape and so we allow $\{Y, N, G\}$ in the image of δ. The symbols Y, N, G are used to classify the final configurations: for a TM solving a certain problem,

- Y will stand for *correct input, acceptance*,
- N for *correct input, rejection*,
- G for *incorrect input*.

Word formalism

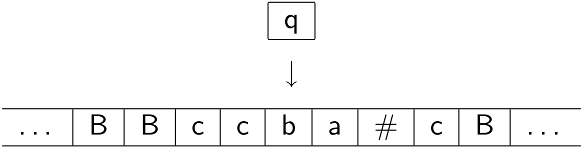
We intend to work exclusively with words, hence we exchange the classical model of “tape” and “control head” by a purely algebraic formalism. Let

$$A' = A \cup \{a^q \mid a \in A, q \in Q\}$$

be the *extended tape alphabet*.

The exponent q on a symbol acnowledges the present scanning of the corresponding cell by the control head, under state q.

Example



$$BBccb^q a\#cB$$

Instantaneous descriptions

Let for : $A'^+ \to A^+$ and exp : $A'^+ \to (\mathbb{N}, +)$ be the "forgetting" and "counting" homomorphisms defined by

$$\mathrm{for}(a) = a, \quad \mathrm{for}(a^q) = a, \quad \exp(a) = 0, \quad \exp(a^q) = 1.$$

We define

$$ID = B^*\{w \in A'^+ \mid \mathrm{for}(w) \in (\{1\} \cup B)(A^o)^*(\{1, \#\}) \\ \cdot (A^o)^*(\{1\} \cup B), \; \exp(w) \; \leq 1\}B^*.$$

$\overline{ID}$ is the set of all nonempty factors of words in ID.

The one-move mapping

The Turing machine $\mathcal{T}$ induces a mapping $\beta : \overline{ID} \to \overline{ID}$ as follows: Let $w \in \overline{ID}$. If $|\exp(w)| = 0$, let $\beta(w) = w$. Suppose now that $w = ua^q v$ with $a \in A$ and $q \in Q$.

- if $\delta(q, a) = b \in \{Y, N, G\}$, let $\beta(w) = ubv$;
- if $\delta(q, a) = (p, b, R)$ and c is the first letter of $v = cv'$, let $\beta(w) = ubc^p v'$;
- if $\delta(q, a) = (p, b, R)$ and $v = 1$, let $\beta(w) = ubB^p$;
- if $\delta(q, a) = (p, b, L)$ and c is the last letter of $u = u'c$, let $\beta(w) = u'c^p bv$;
- if $\delta(q, a) = (p, b, R)$ and $u = 1$, let $\beta(w) = B^p bv$.

Normalized TM

Given $w \in ID$, the sequence $(\beta^n(w))_n$ is eventually constant if and only if $\mathcal{T}$ stops after finitely many moves if and only if $\beta^m(w) \in A^+$ for some $m \in \mathbb{N}$. In this case, we write

$$\lim_{n \to \infty} \beta^n(w) = \beta^m(w).$$

We say that our deterministic Turing machine (TM) is *normalized* if

- $(\beta^n(w))_n$ is eventually constant for every $w \in ID$;
- $\lim_{n \to \infty} \beta^n(w) \in B^* B'^* \{Y, N, G\} B'^* B^*$ for every $w \in ID$.

In view of our stopping conventions, this implies in particular that the symbol Y, N or G must be precisely at the origin.

Space and time

The *space* and *time* functions for the normalized TM $\mathcal{T}$ can be naturally defined by

$$\begin{aligned} s_{\mathcal{T}} : ID &\longrightarrow \mathbb{N} \\ w &\longmapsto |\lim_{n\to\infty} \beta^n(w)|, \end{aligned}$$

$$\begin{aligned} t_{\mathcal{T}} : ID &\longrightarrow \mathbb{N} \\ w &\longmapsto \min\{m \in \mathbb{N} : \beta^m(w) = \lim_{n\to\infty} \beta^n(w)\}. \end{aligned}$$

Any deterministic (multi-tape) Turing machine solving a problem with space and time complexities of order $s(n)$ and $t(n)$ (not less than linear) can be turned into a normalized TM with space and time functions of order $s(n)$ and $(t(n))^2$, respectively.

The one-move lp-mapping

$\beta_0 : ID \to ID$ is defined by

$$\beta_0(w) = \begin{cases} \beta(w) & \text{if } |\beta(w)| = |w| \\ w' & \text{if } |\beta(w)| = |w| + 1 \text{ and } \beta(w) = w'B^p; \\ w' & \text{if } |\beta(w)| = |w| + 1 \text{ and } \beta(w) = B^p w'. \end{cases}$$

Alternatively, we can say that $\beta_0(w)$ is obtained from $\beta(BwB)$ by removing the first and the last letter.

We can deduce $\beta(w)$ from $\beta_0(BwB)$ and, more generally, $\beta^n(w)$ from $\beta_0(B^n w B^n)$.

Space and time from β_0

Let $\iota_B : ID \to (A' \setminus \{B\})^+$ be the mapping that removes all blanks from a given $w \in ID$.

Proposition. Let $\mathcal{T}$ be a normalized TM with one-move mapping β.Let $w \in \overline{ID}$ be such that $\text{for}(w) \in (A \setminus \{B\})^+$. Then

(i) $\lim_{n\to\infty} \beta^n(w) = \lim_{n\to\infty} \iota_B(\beta_0^n(B^n w B^n))$;

(ii) $s_{\mathcal{T}}(w) = |\lim_{n\to\infty} \iota_B(\beta_0^n(B^n w B^n))|$;

(iii) $t_{\mathcal{T}}(w) = \min\{m \in \mathbb{N} : \iota_B(\beta_0^m(B^m w B^m)) = \lim_{n\to\infty} \iota_B(\beta_0^n(B^n w B^n))\}$.

Bad words

We extend $\beta_0 : \overline{ID} \to \overline{ID}$ to an lp-mapping $\beta_0 : A'^+ \to \overline{ID}$ by composing β_0 with $\Delta : A'^+ \to ID$ defined by

$$\Delta(w) = \begin{cases} w & \text{if } w \in \overline{ID} \\ GB'^{|w|-1} & \text{otherwise.} \end{cases}$$

So far, we have associated to the normalized TM $\mathcal{T}$ an lp-mapping β_0 encoding the full computational power of $\mathcal{T}$ with space and time functions equivalent to those of $\mathcal{T}$.

We define next a canonical finite bimachine matching β_0 in $\overline{ID}$.

The one-move bimachine

The A', A'-bimachine

$$\mathcal{B}_{\mathcal{T}} = ((I_R, Q_R, S_R), f, (S_L, Q_L, I_L))$$

is defined as follows:

- $Q_R = A' \cup \{I_R\}$, $Q_L = A' \cup \{I_L\}$;
- $S_R = A'$ is a right zero semigroup ($ab = b$);
- $S_L = A'$ is a left zero semigroup ($ab = a$);
- the action $Q_R \times S_R \to Q_R$ is defined by $q_R a = a$;
- the action $S_L \times Q_L \to Q_L$ is defined by $a q_L = a$

The one-move bimachine

For the output function, let us write $I'_R = B$ and $q'_R = q_R$ for every $q_R \in Q_R \setminus \{I_R\}$. Similarly, we define q'_L. Given $q_R \in Q_R$, $a \in A'$ and $q_L \in Q_L$, let

$$f(q_R, a, q_L) = \beta_0(q'_R, a, q'_L).$$

If $q_R a q_L \in \overline{ID}$, then $q_R a q_L$ will encode the situation of three consecutive tape cells at a certain moment. Then $f(q_R, a, q_L)$ describes the situation of the middle cell after one move of $\mathcal{T}$.

Proposition. Let $\mathcal{T}$ be a normalized TM with one-move lp-mapping β_0. Then $\alpha_{\mathcal{B}_{\mathcal{T}}}(w) = \beta_0(w)$ for every $w \in \overline{ID}$.