

Pebble transducers with unary output

Gaëtan Douéneau-Tabot (IRIF, Université de Paris)

doueneau@irif.fr

Paris, June 29th, 2021



**INSTITUT
DE RECHERCHE
EN INFORMATIQUE
FONDAMENTALE**

Motivation: transducers are simple programs

Finite automata

- Model simple programs that accept/reject: $A^* \rightarrow \{T, \perp\}$
 - ▶ All models (two-way, one-way, etc.) are equivalent
 - ▶ "Everything" is decidable: equivalence, membership, etc.

Deterministic transducers = finite automata + outputs

- Model simple programs that produce an output: $A^* \rightarrow B^*$
 - ▶ Different models have **different expressive power**
 - ▶ Many questions are **more complex**

Class membership problems

- ▶ Decide if a function described by a transducer from a class \mathcal{C} can be computed by a transducer in some "simpler" class \mathcal{C}' .
- ▶ It is program optimization!

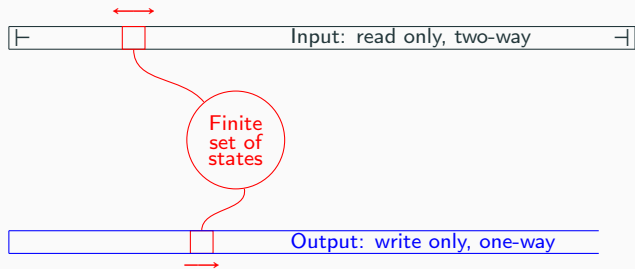
We study membership problems for **pebble transducers**

1. Regular and polyregular functions
2. Two variants of pebble transducers
3. Pebble transducers with unary output

Regular and polyregular functions

Deterministic two-way transducers

Two-way transducer = two-way automaton + outputs



→ Computes a function $f : A^* \rightarrow B^*$

Regular functions

Regular functions = functions computed by two-way transducers

Example: regular functions

- ▶ $\text{remove-a} : \text{abaacdaaa} \mapsto \text{bcd}$
- ▶ $\text{reverse} : \text{abcd} \mapsto \text{dcba}$ (right-to-left reading, detect the borders)

Some properties of regular functions

- ✓ closure under composition [Chytil and Jákl, 1977]
- ✓ logical specifications (MSO-transductions)
[Engelfriet and Hoogeboom, 2001]
- ✓ equivalent to copyless streaming string transducers
[Alur and Cerný, 2010]
- ✓ decidable equivalence problem for two-way transducers [Gurari, 1982]

Deterministic pebble transducers

Ingredients: nested two-way transducers + pebbles

Definition: 0-pebble transducer = two-way transducer

Definition: 1-pebble transducer



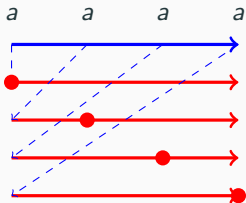
→ k -pebble: stack of submachines of depth k

Polyregular functions

Polyregular functions = functions computed by k -pebble transducers
(for some $k \geq 0$) With $k = 0$ it contains regular functions.

Example: functions computable by 1-pebble

- ▶ u-square : $a^n \mapsto a^{n^2}$
- ▶ #-square : $aaaa \mapsto aaaa\#aaaa\#aaaa\#aaaa\#$
- ▶ #-pref : $aaaa \mapsto a\#aa\#aa\#aaaa\#$
- ▶ square : $aaaa \mapsto \underline{a}aaa\#\underline{a}aa\#\underline{a}aa\#\underline{a}aaa\#$

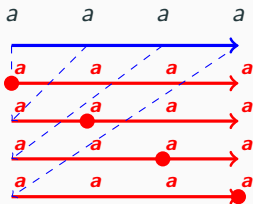


Polyregular functions

Polyregular functions = functions computed by k -pebble transducers
(for some $k \geq 0$) With $k = 0$ it contains regular functions.

Example: functions computable by 1-pebble

- ▶ u-square : $a^n \mapsto a^{n^2}$
- ▶ #-square : $aaaa \mapsto aaaa\#aaaa\#aaaa\#aaaa\#$
- ▶ #-pref : $aaaa \mapsto a\#aa\#aaa\#aaaa\#$
- ▶ square : $aaaa \mapsto \underline{a}aaa\#\underline{a}aa\#\underline{a}aa\#\underline{a}aa\#\underline{a}a\#$

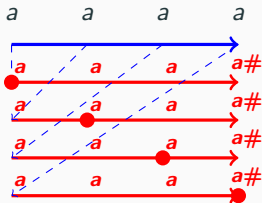


Polyregular functions

Polyregular functions = functions computed by k -pebble transducers
(for some $k \geq 0$) With $k = 0$ it contains regular functions.

Example: functions computable by 1-pebble

- ▶ u-square : $a^n \mapsto a^{n^2}$
- ▶ #-square : $aaaa \mapsto aaaa\#aaaa\#aaaa\#aaaa\#$
- ▶ #-pref : $aaaa \mapsto a\#aa\#aaa\#aaaa\#$
- ▶ square : $aaaa \mapsto \underline{a}aaa\#\underline{a}aa\#\underline{a}aa\#\underline{a}aa\#$

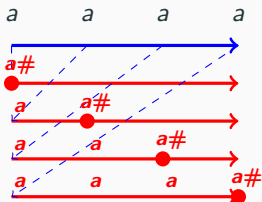


Polyregular functions

Polyregular functions = functions computed by k -pebble transducers
(for some $k \geq 0$) With $k = 0$ it contains regular functions.

Example: functions computable by 1-pebble

- ▶ u-square : $a^n \mapsto a^{n^2}$
- ▶ #-square : $aaaa \mapsto aaaa\#aaaa\#aaaa\#aaaa\#$
- ▶ #-pref : $aaaa \mapsto a\#aa\#aa\#aaaa\#$
- ▶ square : $aaaa \mapsto \underline{a}aaa\#\underline{a}aa\#\underline{a}aa\#\underline{a}aaa\#$

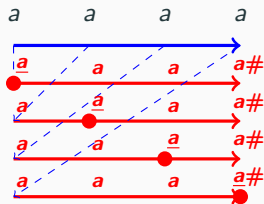


Polyregular functions

Polyregular functions = functions computed by k -pebble transducers
(for some $k \geq 0$) With $k = 0$ it contains regular functions.

Example: functions computable by 1-pebble

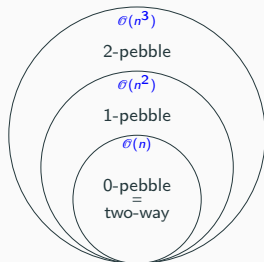
- ▶ u-square : $a^n \mapsto a^{n^2}$
- ▶ #-square : $aaaa \mapsto aaaa\#aaaa\#aaaa\#aaaa\#$
- ▶ #-pref : $aaaa \mapsto a\#aa\#aa\#aaaa\#$
- ▶ square : $aaaa \mapsto \underline{a}aaa\#\underline{a}aa\#\underline{a}aa\#\underline{a}aa\#$



Polyregular functions

Main results on polyregular functions

- ✓ closure under composition [Engelfriet, 2015]
- ✓ logical specifications (MSO-interpretations) [Bojańczyk et al., 2019]
- ✓ "pebble optimisation" (computing the minimal number of pebbles) [Lhote, 2020]

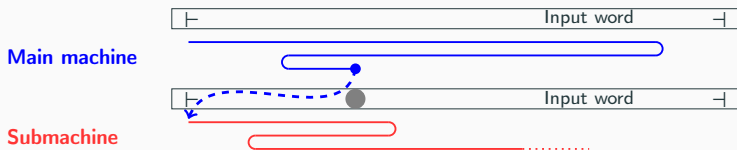


?? equivalence problem of pebble transducers ??

Two variants of pebble transducers

Variants of pebble transducers

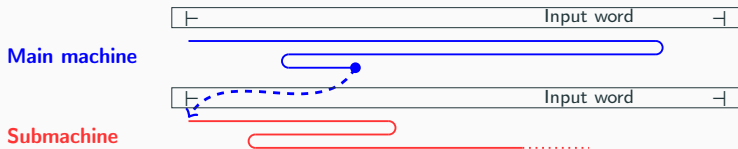
Reminder: 1-pebble transducer



Why have variants been studied?

- ▶ natural restrictions of pebble transducers;
- ▶ equivalence with other models
(register transducers [Douéneau-Tabot et al., 2020], λ -calculus [Nguyễn et al., 2021], etc.)

Definition: 1-blind transducer

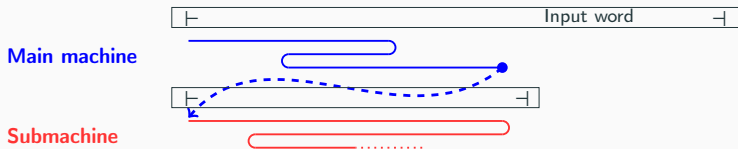


→ k -blind transducer = k -pebble transducer **without pebbles**

Subclasses

- ✓ a k -blind transducer can be simulated by a k -pebble transducer

Definition: 1-marble transducer



→ the submachine works on a **prefix of the input**

Subclasses

- ✓ a k -marble transducer can be simulated by a k -pebble transducer

Strict inclusions for $k = 1$?

Classes of functions [Douéneau-Tabot et al., 2020] [Nguyễn et al., 2021]



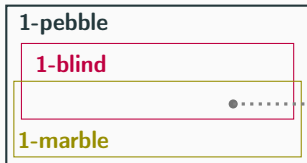
Example: u-square: $a^n \mapsto a^{n^2}$

► Computable by 1-blind?

► Computable by 1-marble?

Strict inclusions for $k = 1$?

Classes of functions [Douéneau-Tabot et al., 2020] [Nguyen et al., 2021]



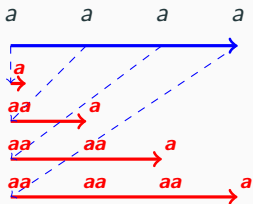
..... u-square: $a^n \mapsto a^{n^2}$

Example: u-square: $a^n \mapsto a^{n^2}$

► Computable by 1-blind?

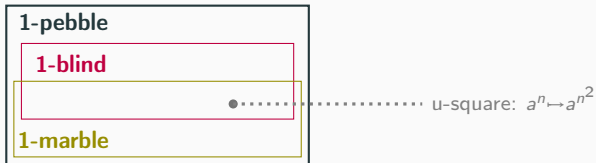


► Computable by 1-marble?



Strict inclusions for $k = 1$?

Classes of functions [Douéneau-Tabot et al., 2020] [Nguyễn et al., 2021]



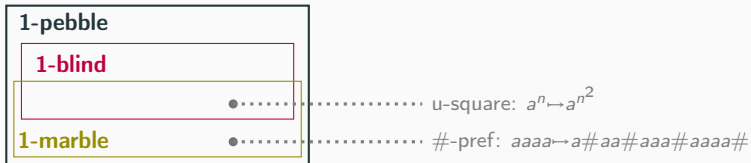
Example: $\#$ -pref : $aaaa \mapsto a\#aa\#aaa\#aaaa\#$

▶ Computable by 1-blind?

▶ Computable by 1-marble?

Strict inclusions for $k = 1$?

Classes of functions [Douéneau-Tabot et al., 2020] [Nguyen et al., 2021]

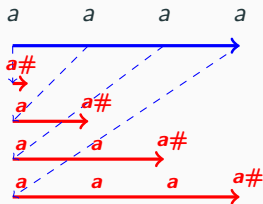


Example: #-pref: $aaaa \mapsto a#aa#aaa#aaaa\#$

► Computable by 1-blind?

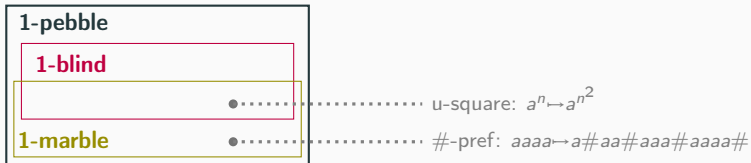
► Computable by 1-marble?

X not computable



Strict inclusions for $k = 1$?

Classes of functions [Douéneau-Tabot et al., 2020] [Nguyễn et al., 2021]



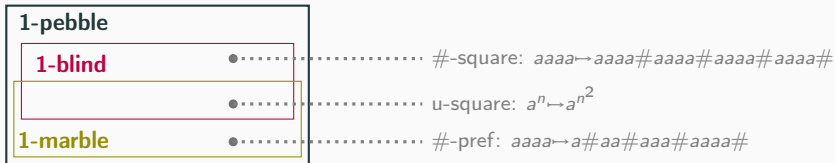
Example: #-square: $aaaa \mapsto aaaa\#aaaa\#aaaa\#aaaa\#$

► Computable by 1-blind?

► Computable by 1-marble?

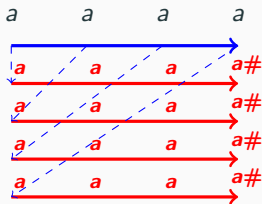
Strict inclusions for $k = 1$?

Classes of functions [Douéneau-Tabot et al., 2020] [Nguyen et al., 2021]



Example: $\#$ -square: $aaaa \mapsto aaaa\#aaaa\#aaaa\#aaaa\#$

► Computable by 1-blind?



► Computable by 1-marble?

X not computable

Strict inclusions for $k = 1$?

Classes of functions [Douéneau-Tabot et al., 2020] [Nguyên et al., 2021]

1-pebble	•.....	square: $aaaa \mapsto \underline{a}aaa\#a\underline{a}a\#aaa\#aaaa\underline{a}\#$
1-blind	•.....	$\#$ -square: $aaaa \mapsto aaaa\#aaaa\#aaaa\#aaaa\#$
	•.....	u-square: $a^n \mapsto a^{n^2}$
1-marble	•.....	$\#$ -pref: $aaaa \mapsto a\#aa\#aaa\#aaaa\#$

Example: square: $aaaa \mapsto \underline{a}aaa\#a\underline{a}a\#aaa\#aaaa\underline{a}\#$

► Computable by 1-blind?

X not computable

► Computable by 1-marble?

X not computable

Pebble transducers with unary output

Transducers with unary output

From now on, machines have a unary output alphabet

In other words $f : A^* \rightarrow \mathbb{N}$

→ Goal: study membership problems

Pebbles and marbles the same

Theorem: k -pebble = k -marble

k -pebble transducers and k -marble transducers with unary output compute the same functions.

(it coincides with rational series of polynomial growth)

!! the conversion may modify the "origin semantics" !!

Corollary:

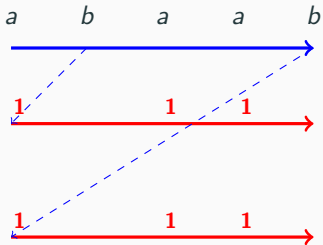
Equivalence of pebble transducers with unary output is decidable.

Pebbles and marbles are the same

Example: u-square : $a^n \mapsto n^2$

Example: prod : $w \in \{a,b\}^* \mapsto |w|_a \times |w|_b$

► 1-pebble



→ for each b , compute the number of a

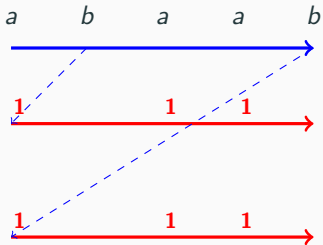
► 1-marble

Pebbles and marbles are the same

Example: u-square : $a^n \mapsto n^2$

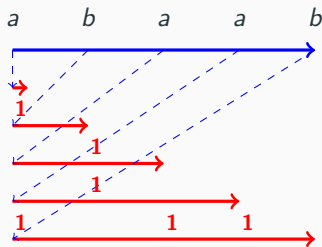
Example: prod : $w \in \{a,b\}^* \mapsto |w|_a \times |w|_b$

► 1-pebble



→ for each b , compute the number of a

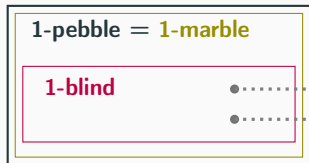
► 1-marble



→ for each b , compute the number of a before it
→ for each a , compute the number of b before it

Strict inclusions for $k = 1$?

Classes of functions with unary output

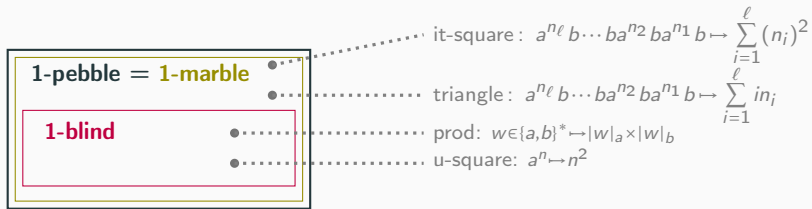


prod: $w \in \{a,b\}^* \mapsto |w|_a \times |w|_b$

u-square: $a^n \mapsto n^2$

Strict inclusions for $k = 1$?

Classes of functions with unary output



Theorem:

One can decide if a function with unary output given by 1-pebble transducer is computable by a 1-blind transducer.

!! the conversion may modify the "origin semantics" !!

Technical tools

- ▶ transition monoids of two-way transducers
- ▶ monoid theory, including **factorization forests**

→ We give a decidable characterization of 1-marble (= 1-pebble) transducers which describe a function computable by 1-blind.

Strict inclusions for $k = 1$

Pumping-like lemma

If f is computable by a k -blind transducer for some $k \geq 0$, then for all $\alpha, \beta \in A^*$, $u \in A^+$, there exists $\omega > 0$ such that if $w = \alpha u^\omega \beta$ then

$$X, Y \mapsto f\left(w^{2\omega-1}(\alpha u^{\omega X} \beta) w^{\omega-1}(\alpha u^{\omega Y} \beta) w^\omega\right)$$

is a polynomial in $X + Y$ for $X, Y \geq 3$.

Strict inclusions for $k = 1$

Pumping-like lemma

If f is computable by a k -blind transducer for some $k \geq 0$, then for all $\alpha, \beta \in A^*$, $u \in A^+$, there exists $\omega > 0$ such that if $w = \alpha u^\omega \beta$ then

$$X, Y \mapsto f\left(w^{2\omega-1}(\alpha u^{\omega X} \beta) w^{\omega-1}(\alpha u^{\omega Y} \beta) w^\omega\right)$$

is a polynomial in $X + Y$ for $X, Y \geq 3$.

Application : it-square: $a^{n_\ell} b \cdots b a^{n_2} b a^{n_1} b \mapsto \sum_{i=1}^{\ell} (n_i)^2$

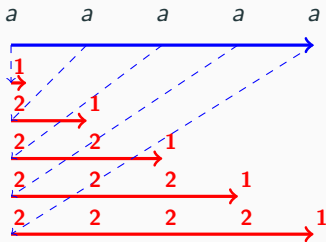
$$\text{it-square}\left((ba^\omega b)^{2\omega-1}(ba^{\omega X} b)(ba^\omega b)^{\omega-1}(ba^{\omega Y} b)(ba^\omega b)^\omega\right)$$

is not a polynomial in $X + Y$ for any $\omega > 0$.

Necessary conditions on "blocks"

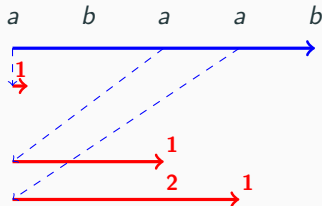
Example 1: two functions computed by 1-marble transducers

► u-square: $a^n \mapsto n^2$



it-square:

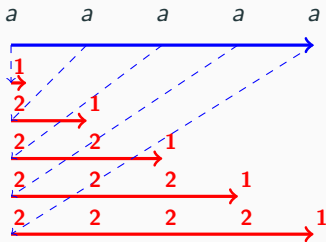
► $a^{n_1} b \dots b a^{n_2} b a^{n_1} b \mapsto \sum_{i=1}^{\ell} (n_i)^2$



Necessary conditions on "blocks"

Example 1: two functions computed by 1-marble transducers

► u-square: $a^n \mapsto n^2$

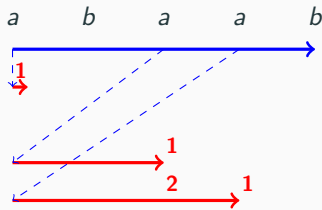


- From a to a : 2

✓ computable by 1-blind

it-square:

► $a^n \ell b \dots b a^{n_2} b a^{n_1} b \mapsto \sum_{i=1}^{\ell} (n_i)^2$



- From a to a :

-> 2 if in same block a^{n_i}

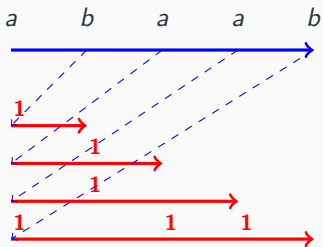
-> 0 otherwise

✗ not k -blind computable

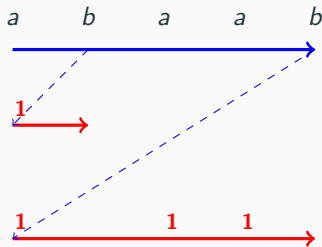
Necessary conditions on "blocks" (2)

Example 2: two functions computed by 1-marble transducers

► $\text{prod} : w \in \{a, b\}^* \mapsto |w|_a \times |w|_b$



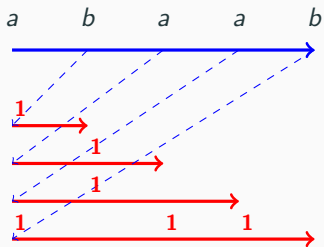
► triangle:
 $a^{n_\ell} b \dots b a^{n_2} b a^{n_1} b \mapsto \sum_{i=1}^{\ell} i n_i$



Necessary conditions on "blocks" (2)

Example 2: two functions computed by 1-marble transducers

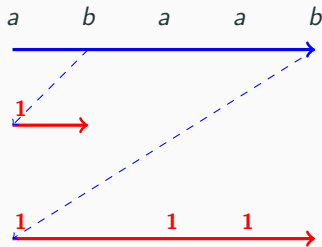
► $\text{prod} : w \in \{a, b\}^* \mapsto |w|_a \times |w|_b$



- From a to b : 1
- From b to a : 1

✓ computable by 1-blind

► triangle:
 $a^{n\ell} b \dots ba^{n2} ba^{n1} b \mapsto \sum_{i=1}^{\ell} in_i$



- From a to b : 0
- From b to a : 1

✗ not k -blind computable

Outlook

New membership techniques

Towards other membership problems:

?? k -pebble transducer with unary output \rightarrow k -blind ??

?? k -pebble transducer (not unary) \rightarrow k -marble ??

\rightarrow contrary to the "pebble optimization" problems of [Lhote, 2020] [Douéneau-Tabot et al., 2020] [Nguyễn et al., 2021] there is no clear intrinsic property of the function for deciding membership

