Transducers beyond linear growth: old and new

Lê Thành Dũng (Tito) Nguyễn — nltd@nguyentito.eu — ÉNS Lyon

Université Libre de Bruxelles, séminaire méthodes formelles – April 18th, 2023

Diversity of transducer theory

Regular languages: a robust notion

deterministic finite automata $\iff$ nondeterministic FA $\iff$ two-way FA $\iff$ regular expressions $\iff$ monadic second-order logic (MSO) $\iff$ ...

How to generalize from *languages* $L \subseteq \Sigma^*$ to *functions* $f\colon \Sigma^* \to \Gamma^*$?

Some equivalences between machine models don't hold anymore with outputs!

e.g. deterministic one-way $\subsetneq$ deterministic two-way (next slide)

Diversity of transducer theory

Regular languages: a robust notion

deterministic finite automata $\iff$ nondeterministic FA $\iff$ two-way FA
$\iff$ regular expressions $\iff$ monadic second-order logic (MSO) $\iff$ ...

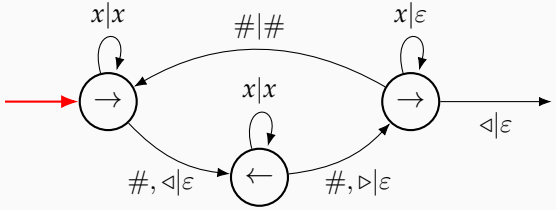
How to generalize from *languages* $L \subseteq \Sigma^*$ to *functions* $f\colon \Sigma^* \to \Gamma^*$?
Some equivalences between machine models don't hold anymore with outputs!
e.g. deterministic one-way $\subsetneq$ deterministic two-way (next slide)

This talk

- Many transducer models (= automata w/ output), some equivalences
- At the end: apply old knowledge (Engelfriet's work)
  to a newer topic (polyregular functions)

Two-way transducers (mentioned in [Shepherdson 1958]!)

Example: $w_1\# \ldots \#w_n \longmapsto w_1 \cdot \texttt{reverse}(w_1)\# \ldots \#w_n \cdot \texttt{reverse}(w_n)$



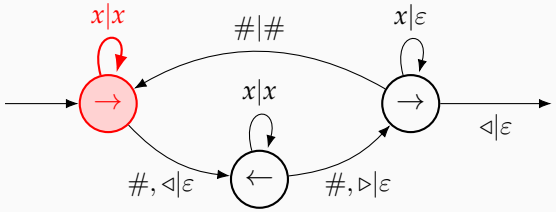
$(x \in \{a, b, c\})$

▷	a	b	c	#	b	a	c	#	c	b	◁

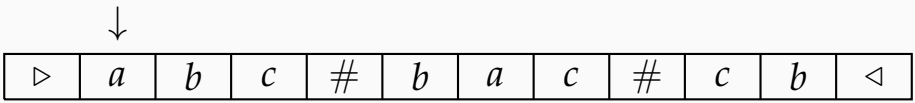
Output:

Two-way transducers (mentioned in [Shepherdson 1958]!)

Example: $w_1\# \ldots \#w_n \longmapsto w_1 \cdot \texttt{reverse}(w_1)\# \ldots \#w_n \cdot \texttt{reverse}(w_n)$



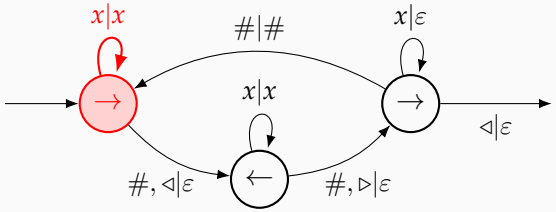
$(x \in \{a, b, c\})$



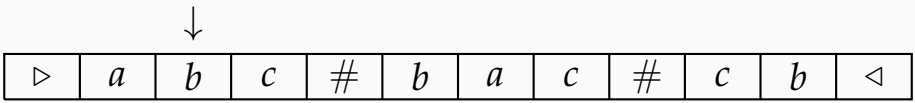
Output:

Two-way transducers (mentioned in [Shepherdson 1958]!)

Example: $w_1\# \ldots \#w_n \longmapsto w_1 \cdot \texttt{reverse}(w_1)\# \ldots \#w_n \cdot \texttt{reverse}(w_n)$



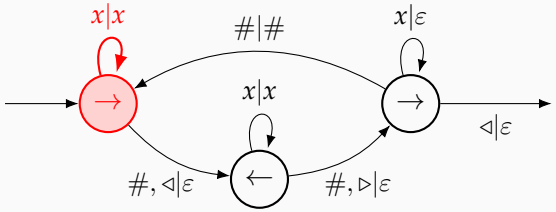
$(x \in \{a, b, c\})$



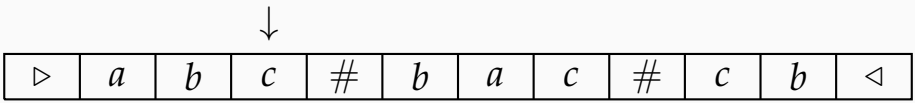
Output: a

Two-way transducers (mentioned in [Shepherdson 1958]!)

Example: $w_1\# \ldots \#w_n \longmapsto w_1 \cdot \texttt{reverse}(w_1)\# \ldots \#w_n \cdot \texttt{reverse}(w_n)$



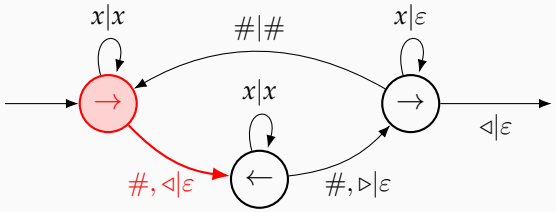
$(x \in \{a, b, c\})$



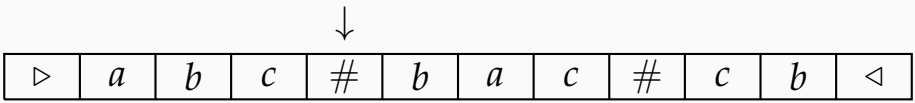
Output: ab

Two-way transducers (mentioned in [Shepherdson 1958]!)

Example: $w_1\# \ldots \#w_n \longmapsto w_1 \cdot \texttt{reverse}(w_1)\# \ldots \#w_n \cdot \texttt{reverse}(w_n)$



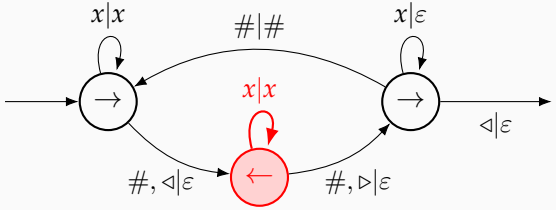
$(x \in \{a, b, c\})$



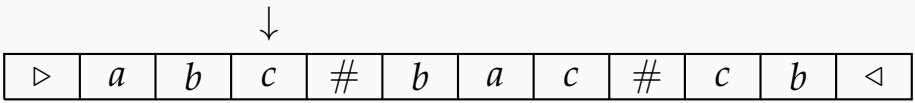
Output: abc

Two-way transducers (mentioned in [Shepherdson 1958]!)

Example: $w_1\# \ldots \#w_n \longmapsto w_1 \cdot \texttt{reverse}(w_1)\# \ldots \#w_n \cdot \texttt{reverse}(w_n)$



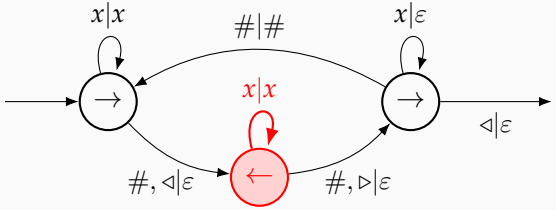
$(x \in \{a, b, c\})$



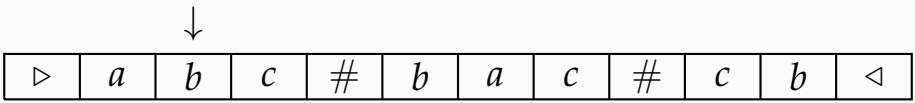
Output: abc

Two-way transducers (mentioned in [Shepherdson 1958]!)

Example: $w_1\# \ldots \#w_n \longmapsto w_1 \cdot \texttt{reverse}(w_1)\# \ldots \#w_n \cdot \texttt{reverse}(w_n)$



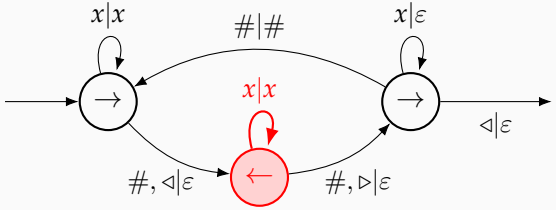
$(x \in \{a, b, c\})$



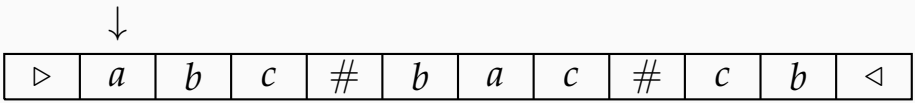
Output: *abcc*

Two-way transducers (mentioned in [Shepherdson 1958]!)

Example: $w_1\# \ldots \#w_n \longmapsto w_1 \cdot \texttt{reverse}(w_1)\# \ldots \#w_n \cdot \texttt{reverse}(w_n)$



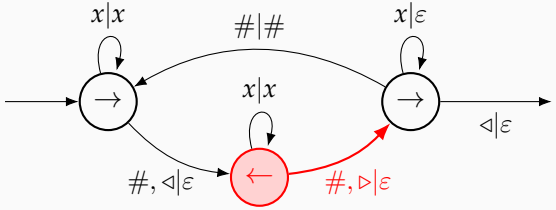
$(x \in \{a, b, c\})$



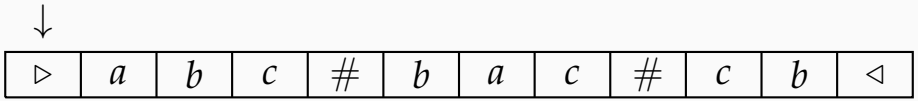
Output: *abccb*

Two-way transducers (mentioned in [Shepherdson 1958]!)

Example: $w_1\# \ldots \#w_n \longmapsto w_1 \cdot \texttt{reverse}(w_1)\# \ldots \#w_n \cdot \texttt{reverse}(w_n)$



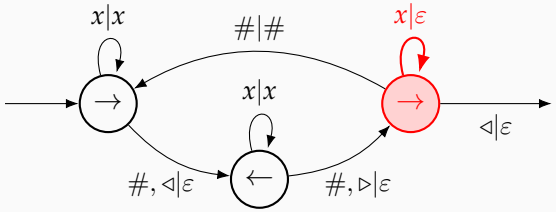
$(x \in \{a, b, c\})$



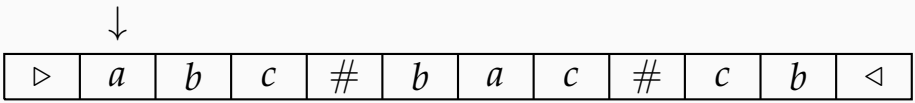
Output: *abccbba*

Two-way transducers (mentioned in [Shepherdson 1958]!)

Example: $w_1\# \ldots \#w_n \longmapsto w_1 \cdot \texttt{reverse}(w_1)\# \ldots \#w_n \cdot \texttt{reverse}(w_n)$



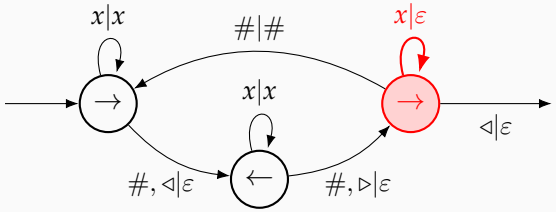
$(x \in \{a, b, c\})$



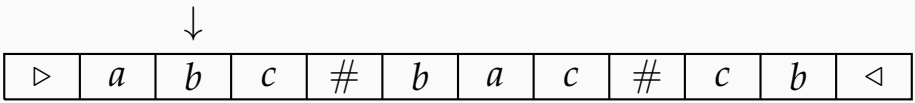
Output: *abccba*

Two-way transducers (mentioned in [Shepherdson 1958]!)

Example: $w_1\# \ldots \#w_n \longmapsto w_1 \cdot \texttt{reverse}(w_1)\# \ldots \#w_n \cdot \texttt{reverse}(w_n)$



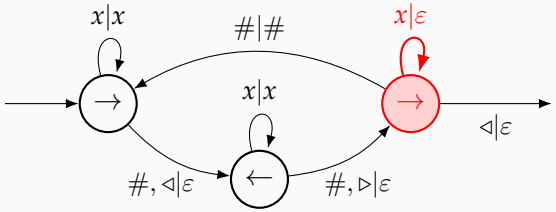
$(x \in \{a, b, c\})$



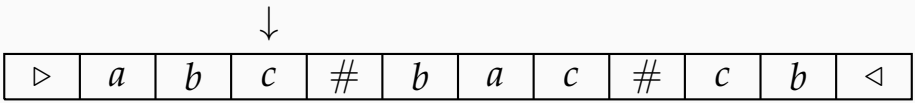
Output: *abccbaa*

Two-way transducers (mentioned in [Shepherdson 1958]!)

Example: $w_1\# \ldots \#w_n \longmapsto w_1 \cdot \texttt{reverse}(w_1)\# \ldots \#w_n \cdot \texttt{reverse}(w_n)$



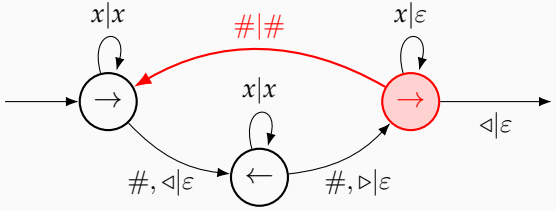
$(x \in \{a, b, c\})$



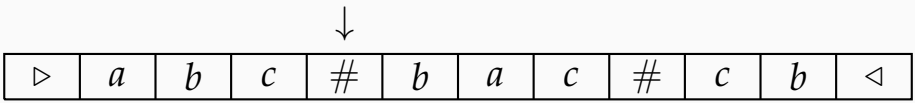
Output: *abccbaa*

Two-way transducers (mentioned in [Shepherdson 1958]!)

Example: $w_1\# \ldots \#w_n \longmapsto w_1 \cdot \texttt{reverse}(w_1)\# \ldots \#w_n \cdot \texttt{reverse}(w_n)$



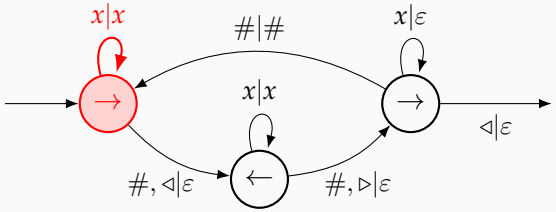
$(x \in \{a, b, c\})$



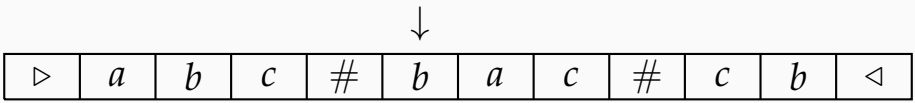
Output: *abccba*

Two-way transducers (mentioned in [Shepherdson 1958]!)

Example: $w_1\# \ldots \#w_n \longmapsto w_1 \cdot \texttt{reverse}(w_1)\# \ldots \#w_n \cdot \texttt{reverse}(w_n)$



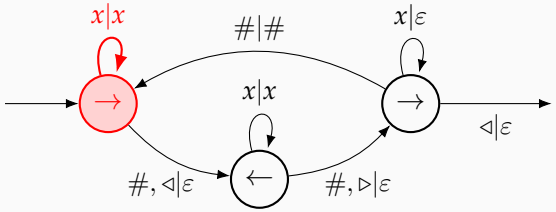
$(x \in \{a, b, c\})$



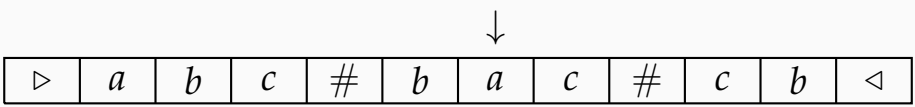
Output: $abccba\#$

Two-way transducers (mentioned in [Shepherdson 1958]!)

Example: $w_1\# \ldots \#w_n \longmapsto w_1 \cdot \texttt{reverse}(w_1)\# \ldots \#w_n \cdot \texttt{reverse}(w_n)$



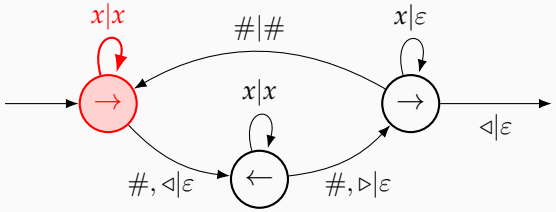
$(x \in \{a, b, c\})$



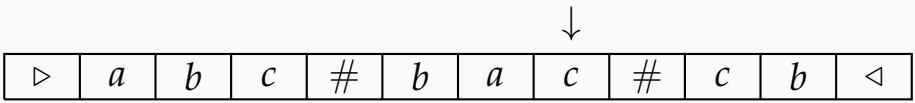
Output: $abccba\#b$

Two-way transducers (mentioned in [Shepherdson 1958]!)

Example: $w_1\# \ldots \#w_n \longmapsto w_1 \cdot \texttt{reverse}(w_1)\# \ldots \#w_n \cdot \texttt{reverse}(w_n)$



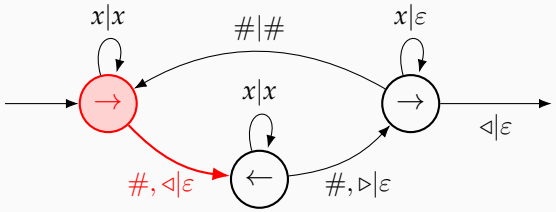
$(x \in \{a, b, c\})$



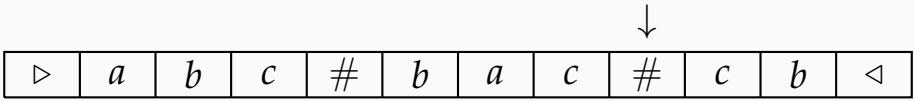
Output: $abccba\#ba$

Two-way transducers (mentioned in [Shepherdson 1958]!)

Example: $w_1\# \dots \# w_n \longmapsto w_1 \cdot \texttt{reverse}(w_1)\# \dots \# w_n \cdot \texttt{reverse}(w_n)$



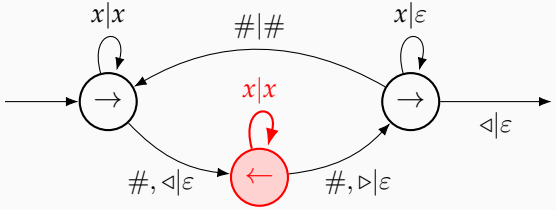
$(x \in \{a, b, c\})$



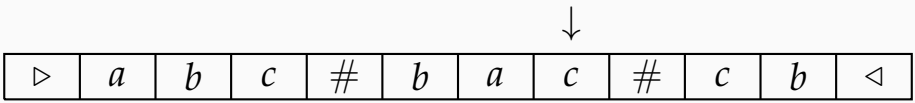
Output: $abccba\#bac$

Two-way transducers (mentioned in [Shepherdson 1958]!)

Example: $w_1\# \ldots \#w_n \longmapsto w_1 \cdot \texttt{reverse}(w_1)\# \ldots \#w_n \cdot \texttt{reverse}(w_n)$



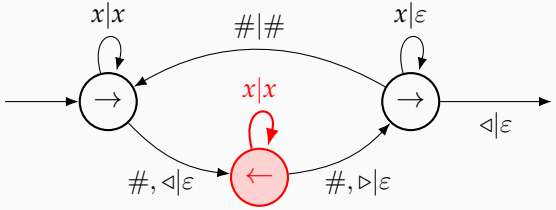
$(x \in \{a, b, c\})$



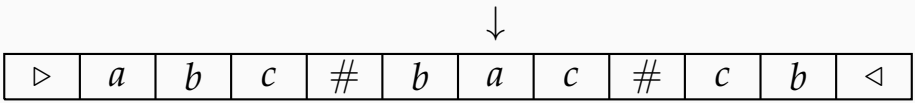
Output: $abccba\#bac$

Two-way transducers (mentioned in [Shepherdson 1958]!)

Example: $w_1\# \dots \# w_n \longmapsto w_1 \cdot \texttt{reverse}(w_1)\# \dots \# w_n \cdot \texttt{reverse}(w_n)$



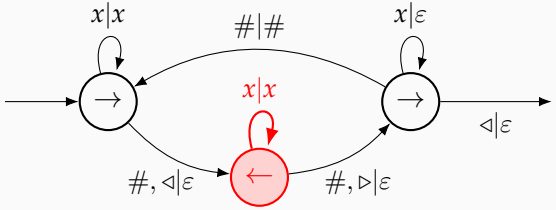
$(x \in \{a, b, c\})$



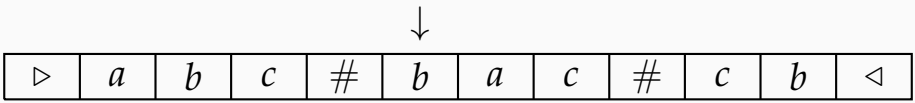
Output: $abccba\#bacc$

Two-way transducers (mentioned in [Shepherdson 1958]!)

Example: $w_1\# \ldots \#w_n \longmapsto w_1 \cdot \texttt{reverse}(w_1)\# \ldots \#w_n \cdot \texttt{reverse}(w_n)$



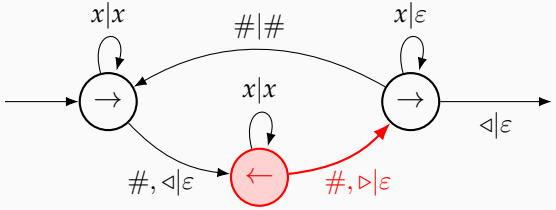
$(x \in \{a, b, c\})$



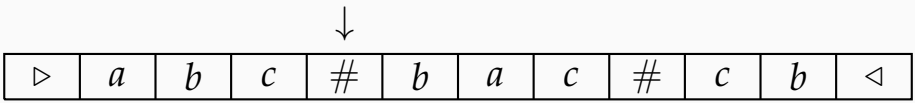
Output: *abccba#bacca*

Two-way transducers (mentioned in [Shepherdson 1958]!)

Example: $w_1\# \ldots \#w_n \longmapsto w_1 \cdot \texttt{reverse}(w_1)\# \ldots \#w_n \cdot \texttt{reverse}(w_n)$



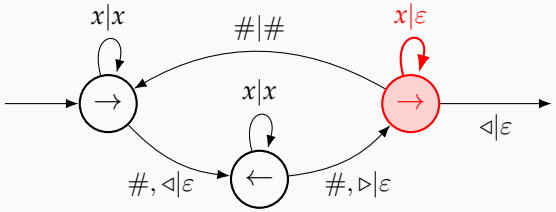
$(x \in \{a, b, c\})$



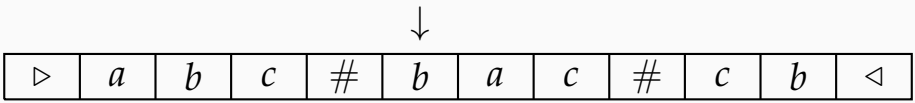
Output: *abccba#baccab*

Two-way transducers (mentioned in [Shepherdson 1958]!)

Example: $w_1\# \ldots \#w_n \longmapsto w_1 \cdot \texttt{reverse}(w_1)\# \ldots \#w_n \cdot \texttt{reverse}(w_n)$



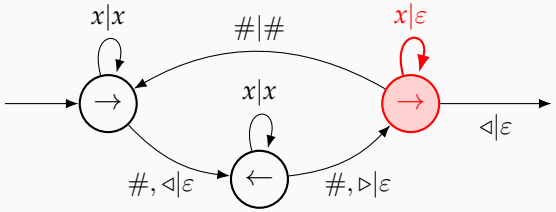
$(x \in \{a, b, c\})$



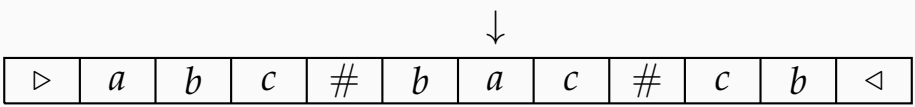
Output: *abccba#baccab*

Two-way transducers (mentioned in [Shepherdson 1958]!)

Example: $w_1\# \ldots \#w_n \longmapsto w_1 \cdot \texttt{reverse}(w_1)\# \ldots \#w_n \cdot \texttt{reverse}(w_n)$



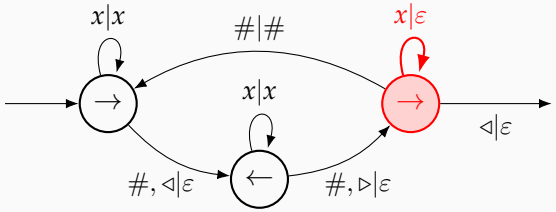
$(x \in \{a, b, c\})$



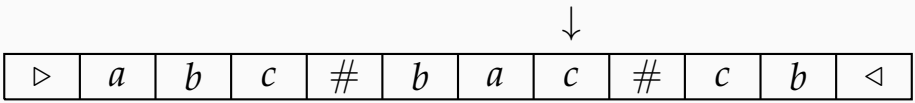
Output: *abccba#baccab*

Two-way transducers (mentioned in [Shepherdson 1958]!)

Example: $w_1\# \dots \#w_n \longmapsto w_1 \cdot \texttt{reverse}(w_1)\# \dots \#w_n \cdot \texttt{reverse}(w_n)$



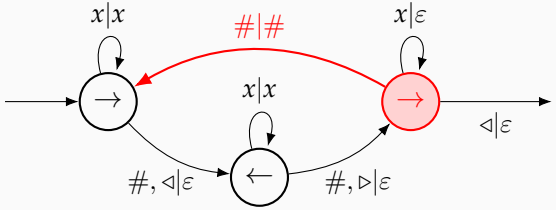
$(x \in \{a, b, c\})$



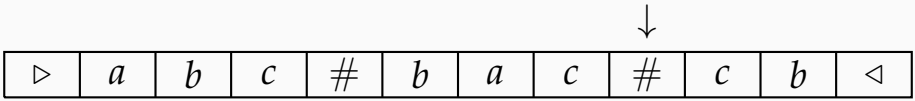
Output: *abccba#baccab*

Two-way transducers (mentioned in [Shepherdson 1958]!)

Example: $w_1\# \ldots \#w_n \longmapsto w_1 \cdot \texttt{reverse}(w_1)\# \ldots \#w_n \cdot \texttt{reverse}(w_n)$



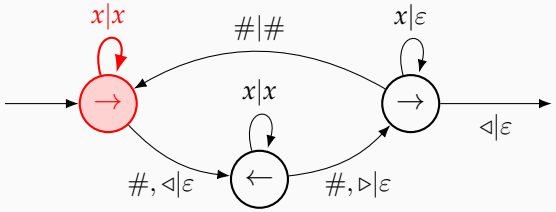
$(x \in \{a, b, c\})$



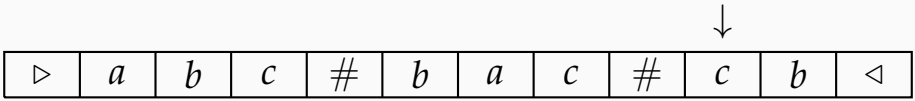
Output: $abccba\#baccab$

Two-way transducers (mentioned in [Shepherdson 1958]!)

Example: $w_1\# \ldots \#w_n \longmapsto w_1 \cdot \texttt{reverse}(w_1)\# \ldots \#w_n \cdot \texttt{reverse}(w_n)$



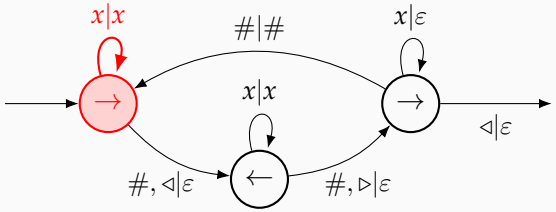
$(x \in \{a, b, c\})$



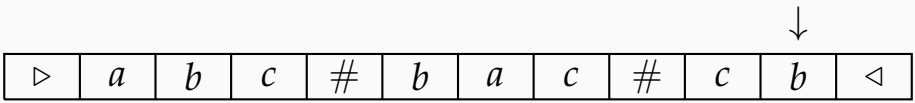
Output: $abccba\#baccab\#$

Two-way transducers (mentioned in [Shepherdson 1958]!)

Example: $w_1\# \ldots \#w_n \longmapsto w_1 \cdot \texttt{reverse}(w_1)\# \ldots \#w_n \cdot \texttt{reverse}(w_n)$



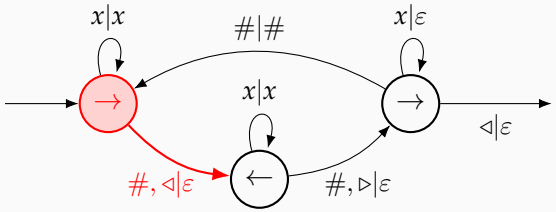
$(x \in \{a, b, c\})$



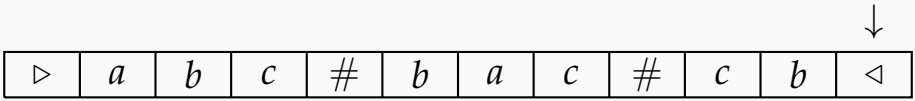
Output: *abccba#baccab#c*

Two-way transducers (mentioned in [Shepherdson 1958]!)

Example: $w_1\# \ldots \#w_n \longmapsto w_1 \cdot \texttt{reverse}(w_1)\# \ldots \#w_n \cdot \texttt{reverse}(w_n)$



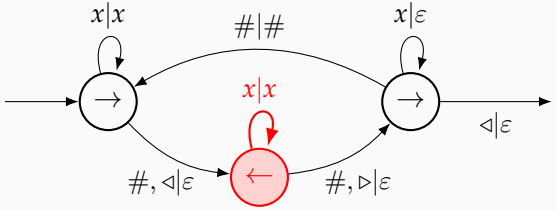
$(x \in \{a, b, c\})$



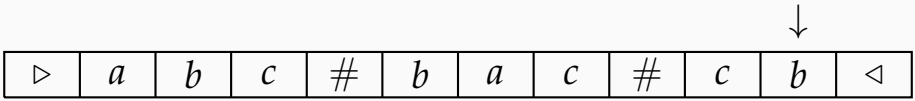
Output: *abccba#baccab#cb*

Two-way transducers (mentioned in [Shepherdson 1958]!)

Example: $w_1\# \ldots \#w_n \longmapsto w_1 \cdot \texttt{reverse}(w_1)\# \ldots \#w_n \cdot \texttt{reverse}(w_n)$



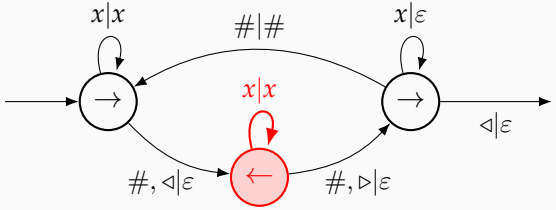
$(x \in \{a, b, c\})$



Output: *abccba#baccab#cb*

Two-way transducers (mentioned in [Shepherdson 1958]!)

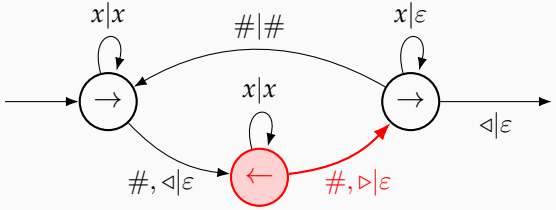
Example: $w_1\# \ldots \#w_n \longmapsto w_1 \cdot \texttt{reverse}(w_1)\# \ldots \#w_n \cdot \texttt{reverse}(w_n)$



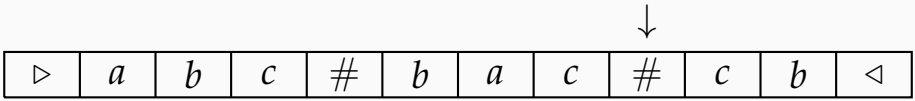
Output: *abccba#baccab#cbb*

Two-way transducers (mentioned in [Shepherdson 1958]!)

Example: $w_1\# \ldots \#w_n \longmapsto w_1 \cdot \texttt{reverse}(w_1)\# \ldots \#w_n \cdot \texttt{reverse}(w_n)$



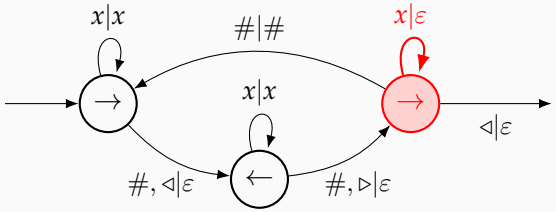
$(x \in \{a, b, c\})$



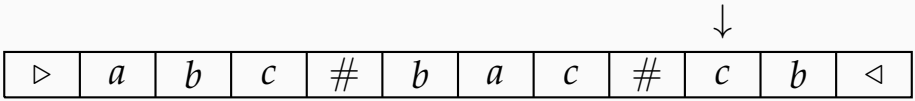
Output: *abccba#baccab#cbbc*

Two-way transducers (mentioned in [Shepherdson 1958]!)

Example: $w_1\# \ldots \#w_n \longmapsto w_1 \cdot \texttt{reverse}(w_1)\# \ldots \#w_n \cdot \texttt{reverse}(w_n)$



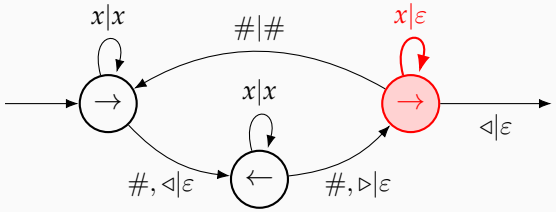
$(x \in \{a, b, c\})$



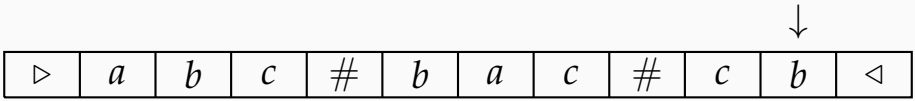
Output: *abccba#baccab#cbbc*

Two-way transducers (mentioned in [Shepherdson 1958]!)

Example: $w_1\# \ldots \#w_n \longmapsto w_1 \cdot \texttt{reverse}(w_1)\# \ldots \#w_n \cdot \texttt{reverse}(w_n)$



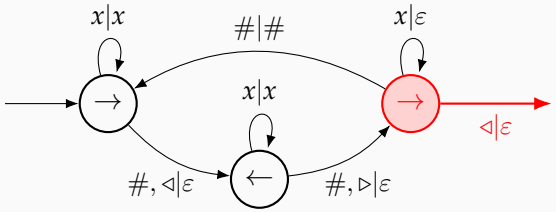
$(x \in \{a, b, c\})$



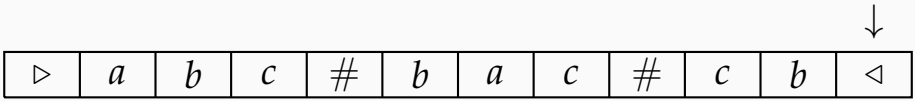
Output: *abccba#baccab#cbbc*

Two-way transducers (mentioned in [Shepherdson 1958]!)

Example: $w_1\# \ldots \#w_n \longmapsto w_1 \cdot \texttt{reverse}(w_1)\# \ldots \#w_n \cdot \texttt{reverse}(w_n)$



$(x \in \{a, b, c\})$



Output: *abccba#baccab#cbbc*

A robust class of linear growth: regular functions

Definition

Regular functions = computed by deterministic two-way transducers

Nice properties: closed under composition; L regular $\implies$ $f^{-1}(L)$ regular

A robust class of linear growth: regular functions

Definition

Regular functions = computed by deterministic two-way transducers

Nice properties: closed under composition; L regular $\implies$ $f^{-1}(L)$ regular

Many equivalent definitions, among others:

- MSO transductions [Engelfriet & Hoogeboom 2001]
- several functional programming characterizations

  e.g. linear λ-calculus [Gallot, Lemay & Salvati 2020; N. & Pradic (in my PhD)]
- copyless streaming string transducers (SSTs) $\longrightarrow$ next slide!

A robust class of linear growth: regular functions

Definition

Regular functions = computed by deterministic two-way transducers

Nice properties: closed under composition; L regular $\implies f^{-1}(L)$ regular

Many equivalent definitions, among others:

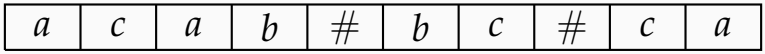
- MSO transductions [Engelfriet & Hoogeboom 2001]
- several functional programming characterizations
  e.g. linear λ-calculus [Gallot, Lemay & Salvati 2020; N. & Pradic (in my PhD)]
- copyless streaming string transducers (SSTs) $\longrightarrow$ next slide!

Well-understood, seems to capture "finite-state computability with linear growth"
$\longrightarrow$ what about functions with $|f(w)| \neq O(|w|)$? our main topic after SSTs

Streaming string transducers [Alur & Černý 2010]

DFA + string-valued *registers*. Example:

$$\begin{array}{lrcl} \texttt{mapReverse}: & \{a,b,c,\#\}^* & \to & \{a,b,c,\#\}^* \\ & w_1\#\ldots\#w_n & \mapsto & \texttt{reverse}(w_1)\#\ldots\#\texttt{reverse}(w_n) \end{array}$$

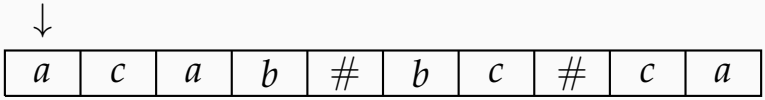


$X = \varepsilon \qquad Y = \varepsilon$

Streaming string transducers [Alur & Černý 2010]

DFA + string-valued *registers*. Example:

$$\texttt{mapReverse}: \quad \{a,b,c,\#\}^* \to \{a,b,c,\#\}^*$$
$$w_1\#\ldots\#w_n \mapsto \texttt{reverse}(w_1)\#\ldots\#\texttt{reverse}(w_n)$$

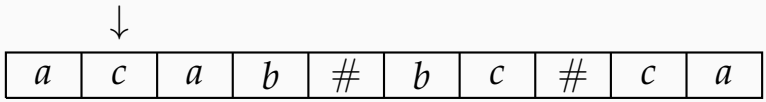


$$X = a \qquad Y = \varepsilon$$

Streaming string transducers [Alur & Černý 2010]

DFA + string-valued *registers*. Example:

$$\texttt{mapReverse}: \quad \{a,b,c,\#\}^* \to \{a,b,c,\#\}^*$$
$$w_1\#\ldots\#w_n \mapsto \texttt{reverse}(w_1)\#\ldots\#\texttt{reverse}(w_n)$$

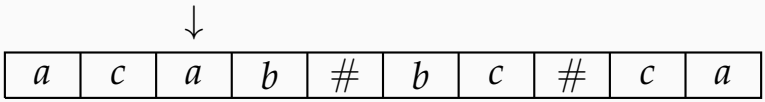


$$X = ca \qquad Y = \varepsilon$$

Streaming string transducers [Alur & Černý 2010]

DFA + string-valued *registers*. Example:

$$\texttt{mapReverse}: \quad \{a,b,c,\#\}^* \to \{a,b,c,\#\}^*$$
$$w_1\#\ldots\#w_n \mapsto \texttt{reverse}(w_1)\#\ldots\#\texttt{reverse}(w_n)$$

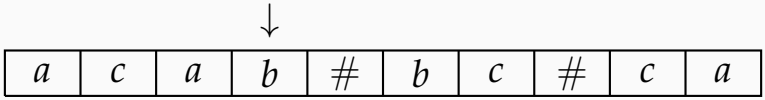


$X = aca \qquad Y = \varepsilon$

Streaming string transducers [Alur & Černý 2010]

DFA + string-valued *registers*. Example:

$$\texttt{mapReverse}: \quad \{a,b,c,\#\}^* \to \{a,b,c,\#\}^*$$
$$w_1\#\ldots\#w_n \mapsto \texttt{reverse}(w_1)\#\ldots\#\texttt{reverse}(w_n)$$

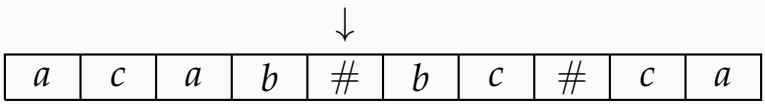


$X = baca \qquad Y = \varepsilon$

Streaming string transducers [Alur & Černý 2010]

DFA + string-valued *registers*. Example:

$$\begin{array}{lccl} \texttt{mapReverse}: & \{a,b,c,\#\}^* & \to & \{a,b,c,\#\}^* \\ & w_1\#\ldots\#w_n & \mapsto & \texttt{reverse}(w_1)\#\ldots\#\texttt{reverse}(w_n) \end{array}$$

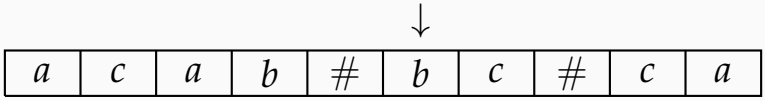


$$X = \varepsilon \qquad Y = baca\#$$

Streaming string transducers [Alur & Černý 2010]

DFA + string-valued *registers*. Example:

$$\texttt{mapReverse}: \quad \{a,b,c,\#\}^* \to \{a,b,c,\#\}^*$$
$$w_1\#\ldots\#w_n \mapsto \texttt{reverse}(w_1)\#\ldots\#\texttt{reverse}(w_n)$$

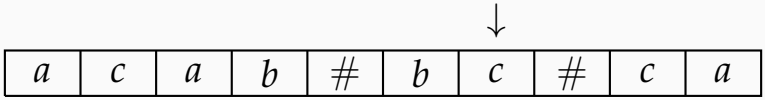


$X = b \qquad Y = baca\#$

Streaming string transducers [Alur & Černý 2010]

DFA + string-valued *registers*. Example:

$$\texttt{mapReverse}: \quad \{a,b,c,\#\}^* \to \{a,b,c,\#\}^*$$
$$w_1\#\ldots\#w_n \mapsto \texttt{reverse}(w_1)\#\ldots\#\texttt{reverse}(w_n)$$

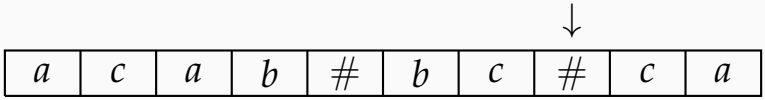


$$X = cb \qquad Y = baca\#$$

Streaming string transducers [Alur & Černý 2010]

DFA + string-valued *registers*. Example:

$$\texttt{mapReverse}: \quad \{a,b,c,\#\}^* \to \{a,b,c,\#\}^*$$
$$w_1\#\ldots\#w_n \mapsto \texttt{reverse}(w_1)\#\ldots\#\texttt{reverse}(w_n)$$

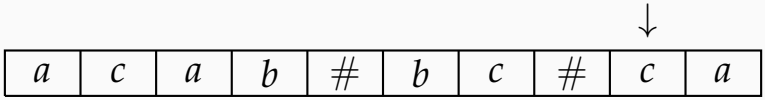


$$X = \varepsilon \qquad Y = baca\#cb\#$$

Streaming string transducers [Alur & Černý 2010]

DFA + string-valued *registers*. Example:

$$\texttt{mapReverse}: \quad \{a,b,c,\#\}^* \to \{a,b,c,\#\}^*$$
$$w_1\#\ldots\#w_n \mapsto \texttt{reverse}(w_1)\#\ldots\#\texttt{reverse}(w_n)$$

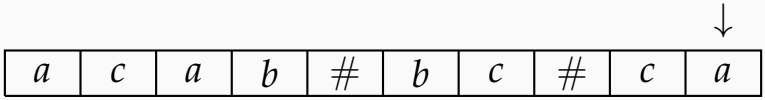


$$X = c \qquad Y = baca\#cb\#$$

Streaming string transducers [Alur & Černý 2010]

DFA + string-valued *registers*. Example:

$$\texttt{mapReverse}: \quad \{a,b,c,\#\}^* \to \{a,b,c,\#\}^*$$
$$w_1\#\ldots\#w_n \mapsto \texttt{reverse}(w_1)\#\ldots\#\texttt{reverse}(w_n)$$

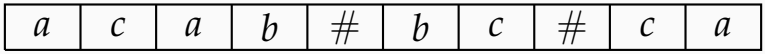


$$X = ac \qquad Y = baca\#cb\#$$

Streaming string transducers [Alur & Černý 2010]

DFA + string-valued *registers*. Example:

$$\begin{array}{lrcl} \texttt{mapReverse}: & \{a,b,c,\#\}^* & \to & \{a,b,c,\#\}^* \\ & w_1\#\ldots\#w_n & \mapsto & \texttt{reverse}(w_1)\#\ldots\#\texttt{reverse}(w_n) \end{array}$$

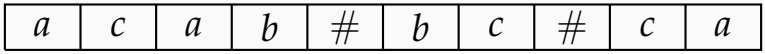


$X = ac \qquad Y = baca\#cb\# \qquad \texttt{mapReverse}(\ldots) = YX = baca\#cb\#ac$

Streaming string transducers [Alur & Černý 2010]

DFA + string-valued *registers*. Example:

$$\begin{array}{lrcl} \texttt{mapReverse}: & \{a,b,c,\#\}^* & \to & \{a,b,c,\#\}^* \\ & w_1\#\ldots\#w_n & \mapsto & \texttt{reverse}(w_1)\#\ldots\#\texttt{reverse}(w_n) \end{array}$$



$X = ac \qquad Y = baca\#cb\# \qquad \texttt{mapReverse}(\ldots) = YX = baca\#cb\#ac$

Regular functions = computed by copyless SSTs

$$a \mapsto \begin{cases} X := aX \\ Y := Y \end{cases} \qquad \# \mapsto \begin{cases} X := \varepsilon \\ Y := YX\# \end{cases}$$

each register appears *at most once* on the right of a $:=$ in a transition

Going beyond linear growth: copyful SSTs

Without the copyless assumption: much more functions computable

- polynomial example: $abc \mapsto (a)(ab)(abc)$ with $a \mapsto \begin{cases} X := Xa \\ Y := YX \end{cases}$

Going beyond linear growth: copyful SSTs

Without the copyless assumption: much more functions computable

- polynomial example: $abc \mapsto (a)(ab)(abc)$ with $a \mapsto \begin{cases} X := Xa \\ Y := YX \end{cases}$
- can grow up to exponentially, e.g. $X := XX$

We still have L regular $\implies f^{-1}(L)$ regular

Going beyond linear growth: copyful SSTs

Without the copyless assumption: much more functions computable

- polynomial example: $abc \mapsto (a)(ab)(abc)$ with $a \mapsto \begin{cases} X := Xa \\ Y := YX \end{cases}$
- can grow up to exponentially, e.g. $X := XX$

We still have L regular $\implies f^{-1}(L)$ regular

Theorem (Filiot & Reynier 2017)

- *The much older* HDT0L systems *are isomorphic to "simple" copyful SSTs*
- *Copyful SSTs can be simplified → they compute* HDT0L transductions

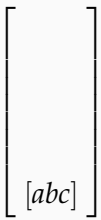
$\longrightarrow$ a function class also studied by [Ferté, Marin & Sénizergues 2014]

Pushdowns of pushdowns

Theorem (Ferté, Marin & Sénizergues 2014)

Right-to-left (simple) copyful SSTs $\iff$ *level-2 pushdown transducers*

Let's compute $abc \mapsto (c)(bc)(abc)$



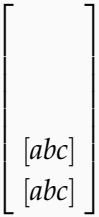
Output:

Pushdowns of pushdowns

Theorem (Ferté, Marin & Sénizergues 2014)

Right-to-left (simple) copyful SSTs $\iff$ *level-2 pushdown transducers*

Let's compute $abc \mapsto (c)(bc)(abc)$



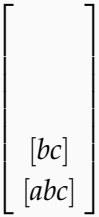
Output:

Pushdowns of pushdowns

Theorem (Ferté, Marin & Sénizergues 2014)

Right-to-left (simple) copyful SSTs $\iff$ *level-2 pushdown transducers*

Let's compute $abc \mapsto (c)(bc)(abc)$



Output:

Pushdowns of pushdowns

Theorem (Ferté, Marin & Sénizergues 2014)

Right-to-left (simple) copyful SSTs $\iff$ *level-2 pushdown transducers*

Let's compute $abc \mapsto (c)(bc)(abc)$

$$\begin{bmatrix} \\ \\ [bc] \\ [bc] \\ [abc] \end{bmatrix}$$

Output:

Pushdowns of pushdowns

Theorem (Ferté, Marin & Sénizergues 2014)

Right-to-left (simple) copyful SSTs $\iff$ *level-2 pushdown transducers*

Let's compute $abc \mapsto (c)(bc)(abc)$

$$\begin{bmatrix} \\ \\ [c] \\ [bc] \\ [abc] \end{bmatrix}$$

Output:

Pushdowns of pushdowns

Theorem (Ferté, Marin & Sénizergues 2014)

Right-to-left (simple) copyful SSTs $\iff$ *level-2 pushdown transducers*

Let's compute $abc \mapsto (c)(bc)(abc)$

$$\begin{bmatrix} \\ [c] \\ [c] \\ [bc] \\ [abc] \end{bmatrix}$$

Output:

Pushdowns of pushdowns

Theorem (Ferté, Marin & Sénizergues 2014)

Right-to-left (simple) copyful SSTs $\iff$ *level-2 pushdown transducers*

Let's compute $abc \mapsto (c)(bc)(abc)$

$$\begin{bmatrix} \\ [] \\ [c] \\ [bc] \\ [abc] \end{bmatrix}$$

Output:

Pushdowns of pushdowns

Theorem (Ferté, Marin & Sénizergues 2014)

Right-to-left (simple) copyful SSTs $\iff$ *level-2 pushdown transducers*

Let's compute $abc \mapsto (c)(bc)(abc)$

$$\begin{bmatrix} \\ \\ [c] \\ [bc] \\ [abc] \end{bmatrix}$$

Output:

Pushdowns of pushdowns

Theorem (Ferté, Marin & Sénizergues 2014)

Right-to-left (simple) copyful SSTs $\iff$ *level-2 pushdown transducers*

Let's compute $abc \mapsto (c)(bc)(abc)$

$$\begin{bmatrix} \\ \\ [] \\ [bc] \\ [abc] \end{bmatrix}$$

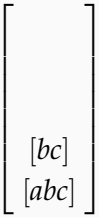
Output: c

Pushdowns of pushdowns

Theorem (Ferté, Marin & Sénizergues 2014)

Right-to-left (simple) copyful SSTs $\iff$ *level-2 pushdown transducers*

Let's compute $abc \mapsto (c)(bc)(abc)$



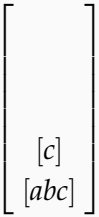
Output: c

Pushdowns of pushdowns

Theorem (Ferté, Marin & Sénizergues 2014)

Right-to-left (simple) copyful SSTs $\iff$ *level-2 pushdown transducers*

Let's compute $abc \mapsto (c)(bc)(abc)$



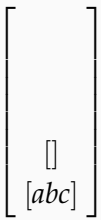
Output: cb

Pushdowns of pushdowns

Theorem (Ferté, Marin & Sénizergues 2014)

Right-to-left (simple) copyful SSTs $\iff$ *level-2 pushdown transducers*

Let's compute $abc \mapsto (c)(bc)(abc)$



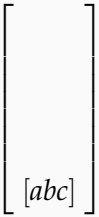
Output: cbc

Pushdowns of pushdowns

Theorem (Ferté, Marin & Sénizergues 2014)

Right-to-left (simple) copyful SSTs $\iff$ *level-2 pushdown transducers*

Let's compute $abc \mapsto (c)(bc)(abc)$



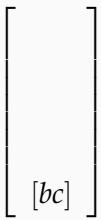
Output: cbc

Pushdowns of pushdowns

Theorem (Ferté, Marin & Sénizergues 2014)

Right-to-left (simple) copyful SSTs $\iff$ *level-2 pushdown transducers*

Let's compute $abc \mapsto (c)(bc)(abc)$



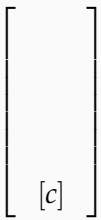
Output: $cbca$

Pushdowns of pushdowns

Theorem (Ferté, Marin & Sénizergues 2014)

Right-to-left (simple) copyful SSTs $\iff$ *level-2 pushdown transducers*

Let's compute $abc \mapsto (c)(bc)(abc)$



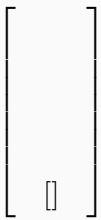
Output: *cbcab*

Pushdowns of pushdowns

Theorem (Ferté, Marin & Sénizergues 2014)

Right-to-left (simple) copyful SSTs $\iff$ *level-2 pushdown transducers*

Let's compute $abc \mapsto (c)(bc)(abc)$



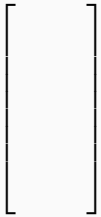
Output: *cbcabc*

Pushdowns of pushdowns

Theorem (Ferté, Marin & Sénizergues 2014)

Right-to-left (simple) copyful SSTs $\iff$ *level-2 pushdown transducers*

Let's compute $abc \mapsto (c)(bc)(abc)$



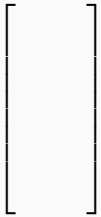
Output: *cbcabc*

Pushdowns of pushdowns

Theorem (Ferté, Marin & Sénizergues 2014)

Right-to-left (simple) copyful SSTs $\iff$ *level-2 pushdown transducers*

Let's compute $abc \mapsto (c)(bc)(abc)$



Output: *cbcabc*

Remark: we never need to push something on the small stacks, they're *input suffixes*

$\longrightarrow$ stack of stack of letters $\simeq$ stack of suffixes

Iterated pushdown transducers: using pushdowns of ... of pushdowns

Claim (Sénizergues 2007 — proof on arXiv for $k \leq 2$)

Composition *of k right-to-left copyful SSTs* (*previous example:* $k = 1$)

$\iff$ *level-*$(k+1)$ *pushdown transducers* (*storing* letters)

$\iff$ *level-k pushdown transducers* (*storing* suffixes)

Composing exponential growth functions $\rightsquigarrow$ towers of exp!

Iterated pushdown transducers: using pushdowns of ... of pushdowns

Claim (Sénizergues 2007 — proof on arXiv for $k \leq 2$)

Composition *of k right-to-left copyful SSTs* (*previous example:* $k = 1$)
$\iff$ *level-*$(k+1)$ *pushdown transducers* (*storing* letters)
$\iff$ *level-k pushdown transducers* (*storing* suffixes)

Composing exponential growth functions $\rightsquigarrow$ towers of exp!

Theorem (Engelfriet & Vogler 1986)

Composition of k macro tree transducers $\iff$ *level-k pushdown transducers manipulating* subtrees *of the input tree*

Macro tree transducers [Engelfriet & Vogler 1985] can be seen as bottom-up automata with registers, generalizing right-to-left copyful SSTs to trees.

“Engelfriet’s class” of transductions

Composition of k macro tree transducers [Engelfriet & Vogler 1986, 1988]

$\iff$ level-k pushdown transducers (manipulating subtrees of the input tree)

$\iff$ “high level tree transducers” of order k

($\simeq$ storing order-k *functions* in registers, with subtle restrictions)

+ the union over all $k \in \mathbb{N}$ = “composition of X” for several other Xs

“Engelfriet’s class” of transductions

Composition of k macro tree transducers [Engelfriet & Vogler 1986, 1988]

$\iff$ level-k pushdown transducers (manipulating subtrees of the input tree)

$\iff$ “high level tree transducers” of order k

($\simeq$ storing order-k *functions* in registers, with subtle restrictions)

+ the union over all $k \in \mathbb{N}$ = “composition of X” for several other Xs

So far, two robust classes closed under composition

- Regular functions: linear growth (& linear-time computable)
- “Engelfriet’s class”: towers of exponentials

What about *polynomial* (growth|time)? (“*feasible* finite-state computation”)

Towards transducers of polynomial growth

Reminder

Regular functions = computed by two-way transducers

$O(|\text{input}|)$ configurations $\{\text{states}\} \times \{\text{reading head positions}\} \implies$ linear growth

Towards transducers of polynomial growth

Reminder

Regular functions = computed by two-way transducers

$O(|\text{input}|)$ configurations $\{\text{states}\} \times \{\text{reading head positions}\} \implies$ linear growth

$\longrightarrow$ to get polynomial growth: multiple heads!

- with no restriction, you get Logspace [Hartmanis 1972]
- idea: impose a "stack condition" on the heads

  $\longrightarrow$ *pebble transducers*, the historical definition of *polyregular functions*

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$

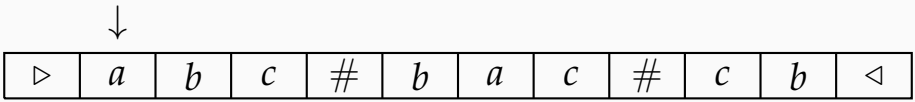
Output:

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



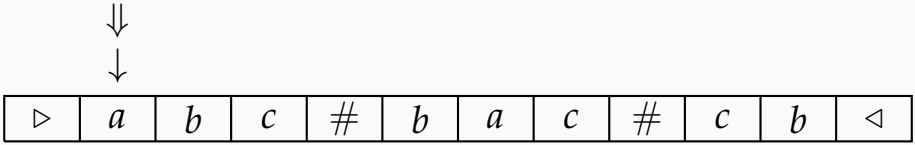
Output:

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



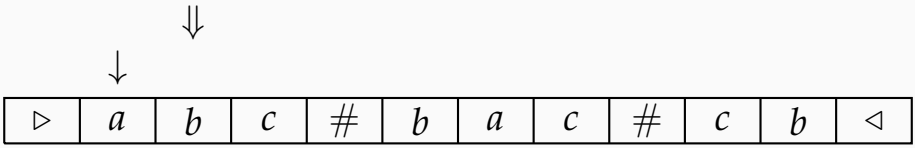
Output:

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



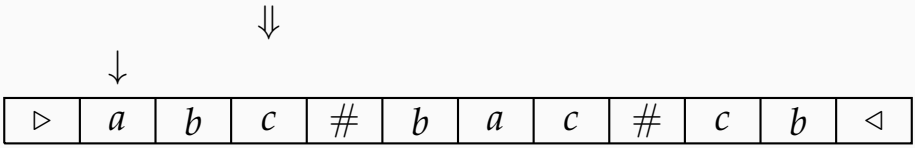
Output:

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



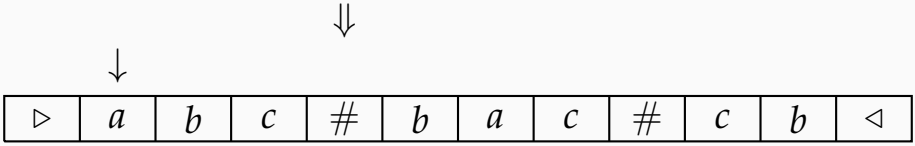
Output:

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



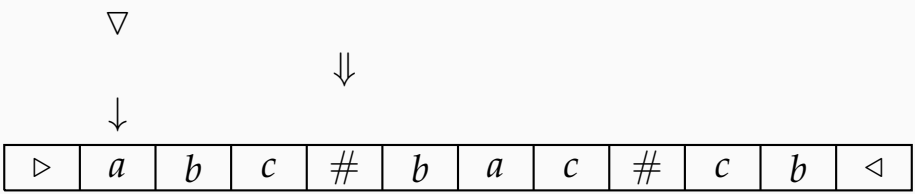
Output:

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



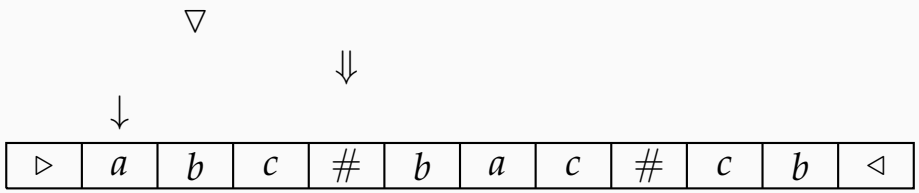
Output:

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



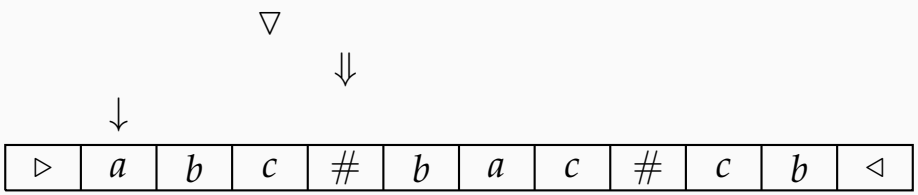
Output: a

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



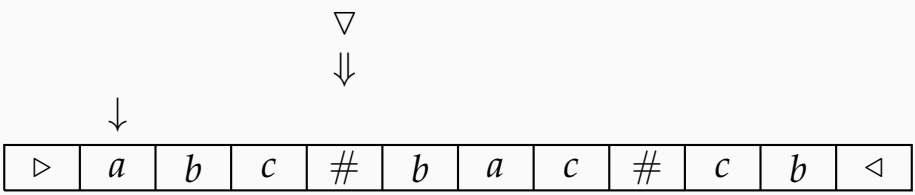
Output: ab

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



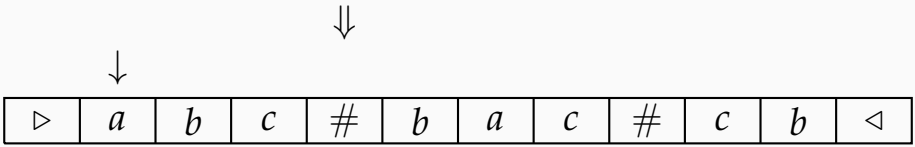
Output: abc

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



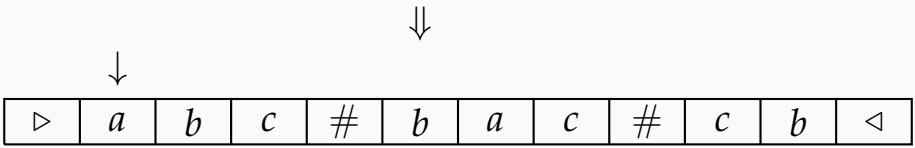
Output: abc

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



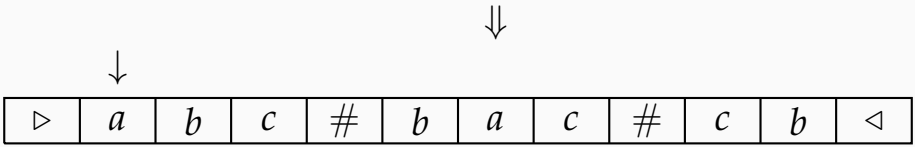
Output: abc

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



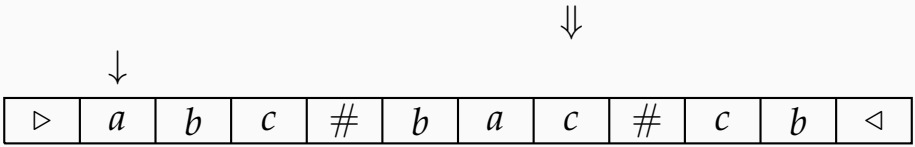
Output: abc

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



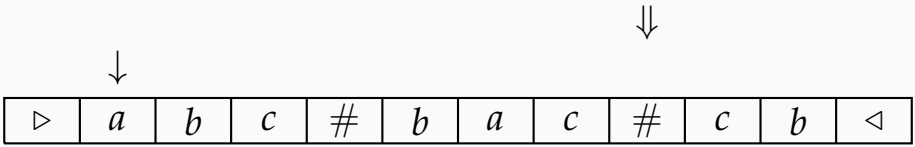
Output: abc

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



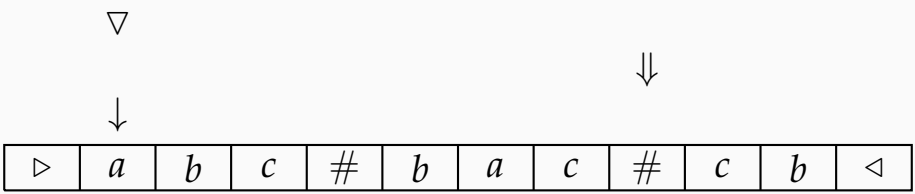
Output: abc

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



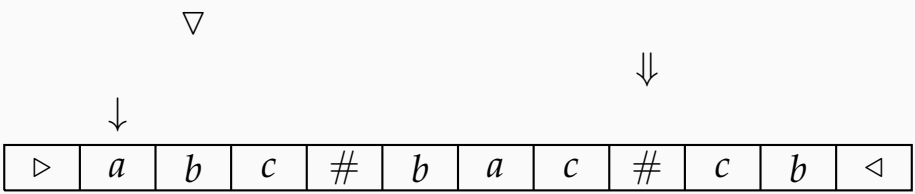
Output: abc

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



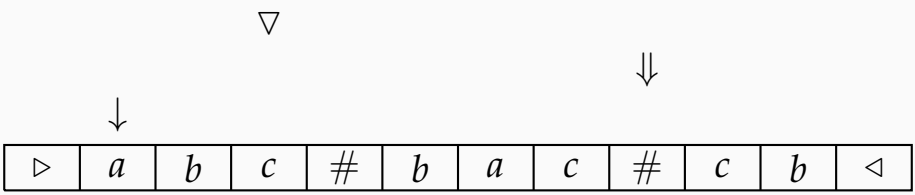
Output: *abca*

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



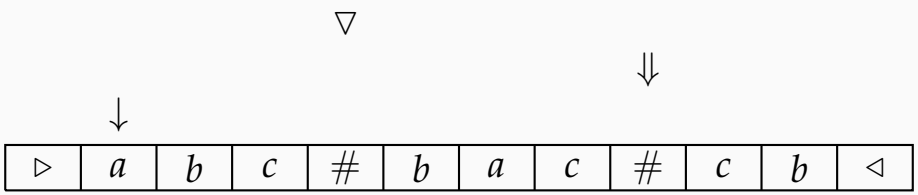
Output: *abcab*

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



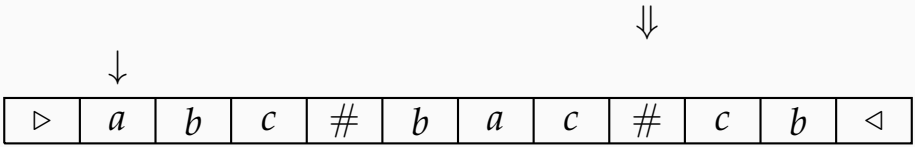
Output: *abcabc*

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



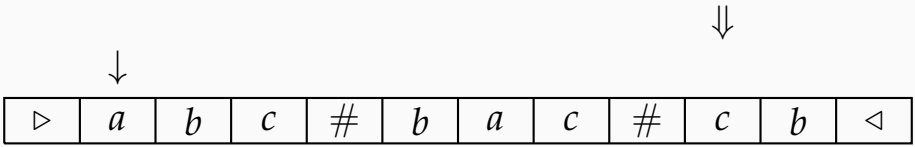
Output: *abcabc*

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



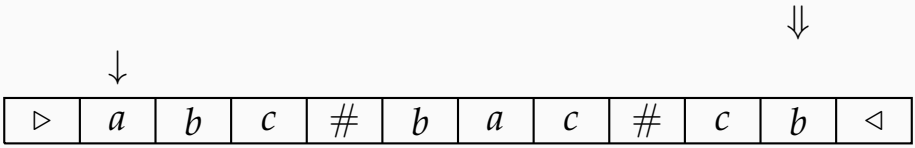
Output: *abcabc*

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



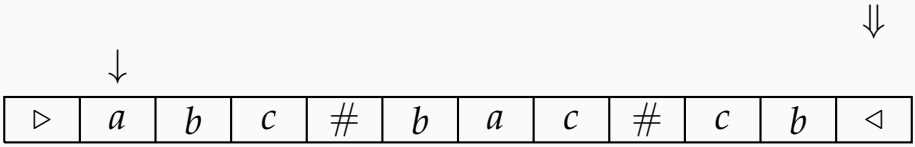
Output: *abcabc*

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



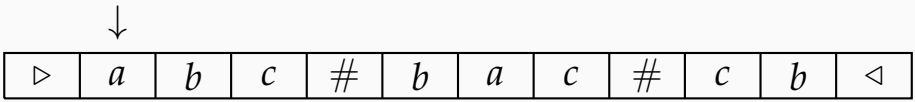
Output: *abcabc*

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



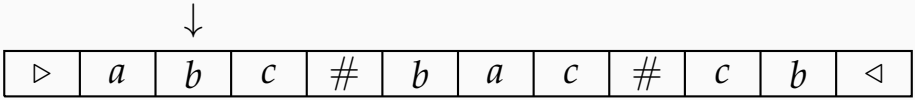
Output: $abcabc\#$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



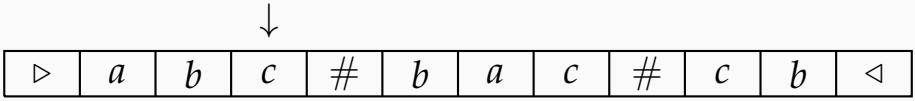
Output: $abcabc\#$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



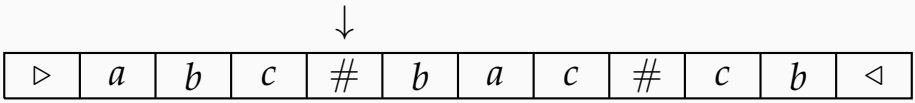
Output: $abcabc\#$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



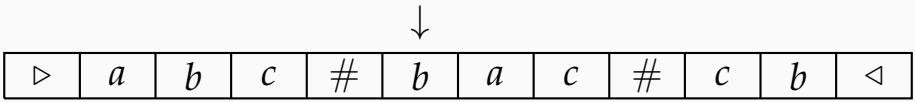
Output: $abcabc\#$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



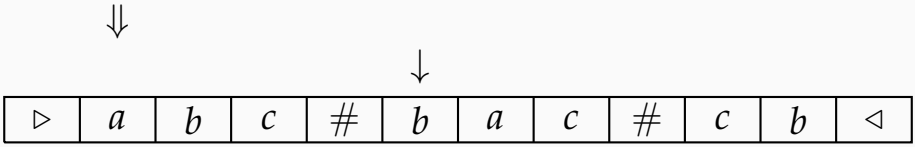
Output: $abcabc\#$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



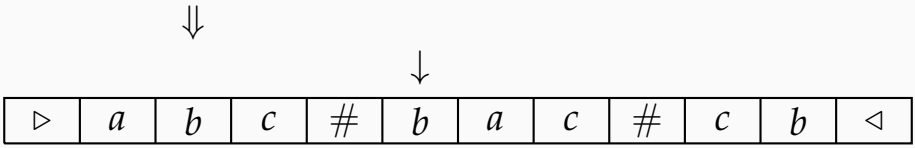
Output: $abcabc\#$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



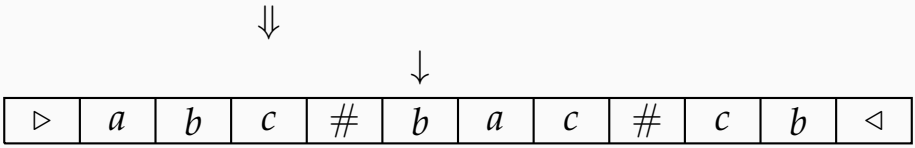
Output: $abcabc\#$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



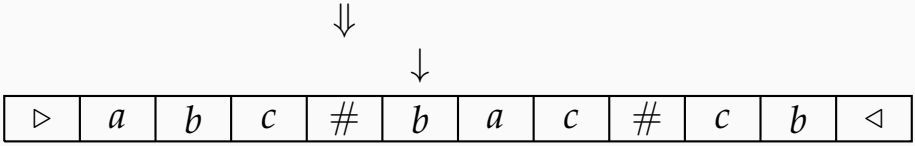
Output: $abcabc\#$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



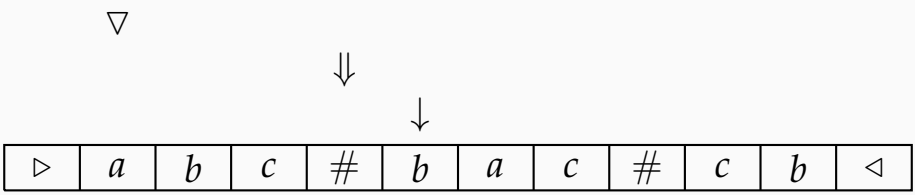
Output: $abcabc\#$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



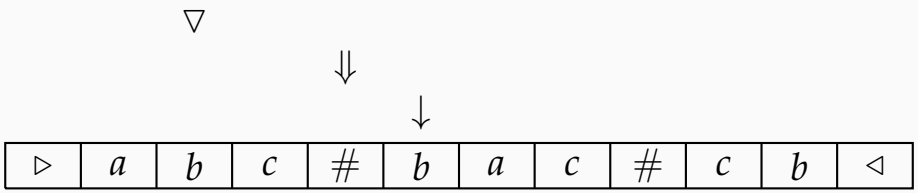
Output: $abcabc\#$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



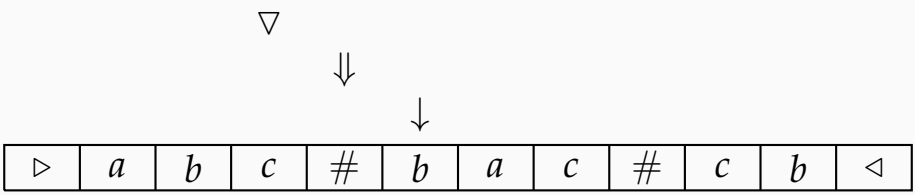
Output: $abcabc\#$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



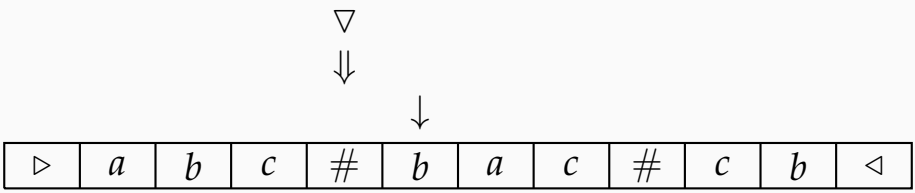
Output: $abcabc\#$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



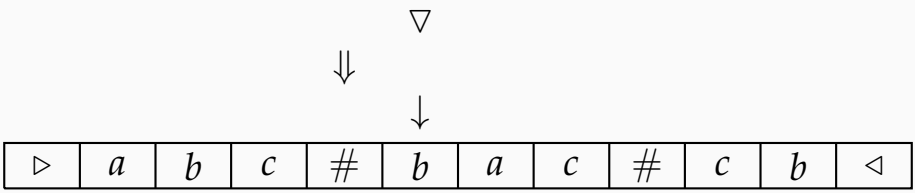
Output: $abcabc\#$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



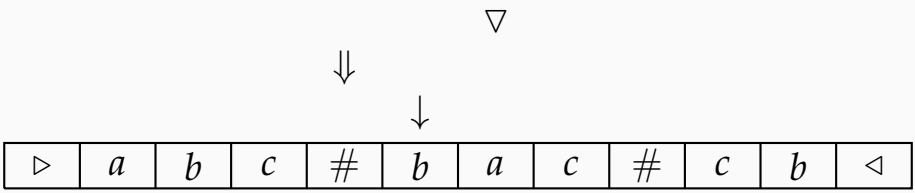
Output: $abcabc\#$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



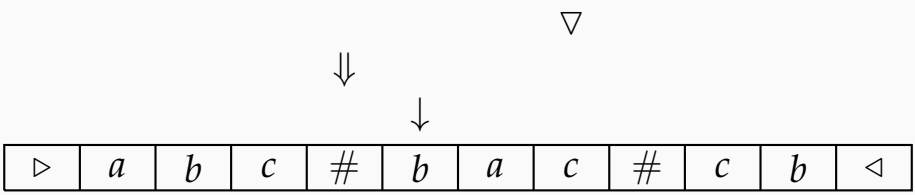
Output: $abcabc\#b$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



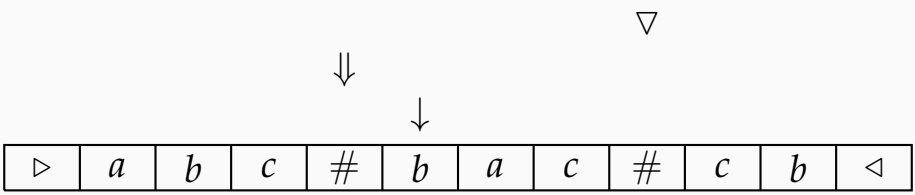
Output: $abcabc\#ba$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



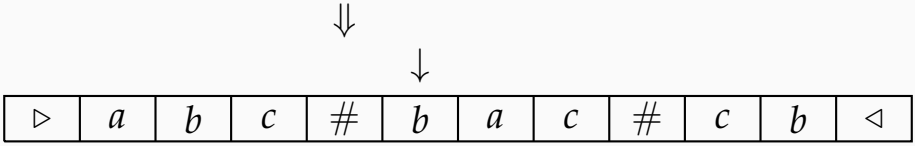
Output: $abcabc\#bac$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



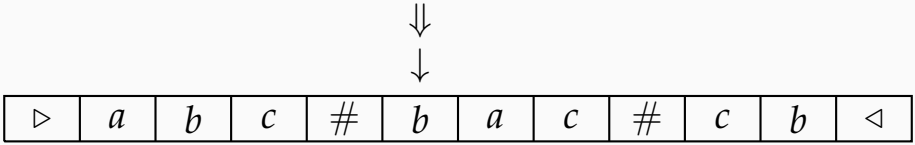
Output: $abcabc\#bac$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



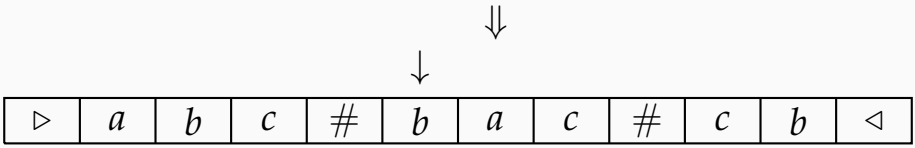
Output: $abcabc\#bac$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



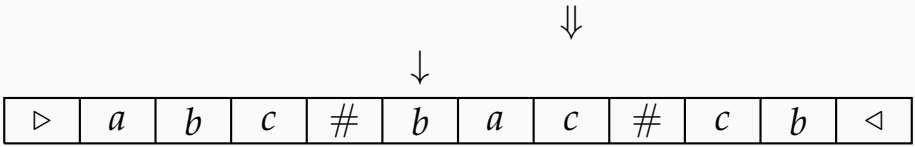
Output: $abcabc\#bac$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



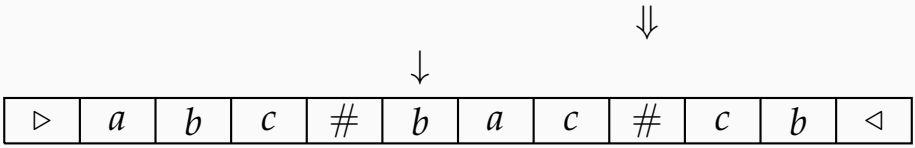
Output: $abcabc\#bac$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



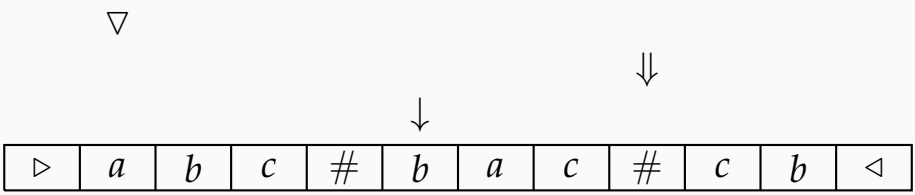
Output: $abcabc\#bac$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



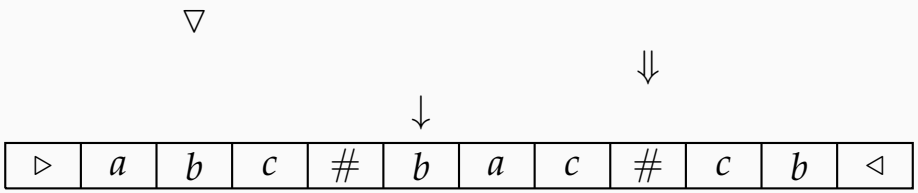
Output: *abcabc#bac*

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



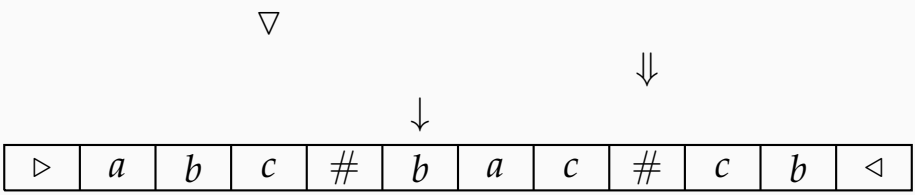
Output: $abcabc\#bac$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



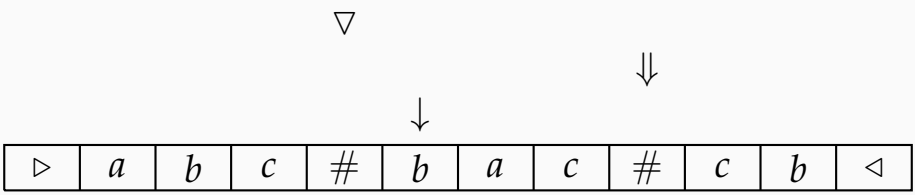
Output: *abcabc#bac*

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



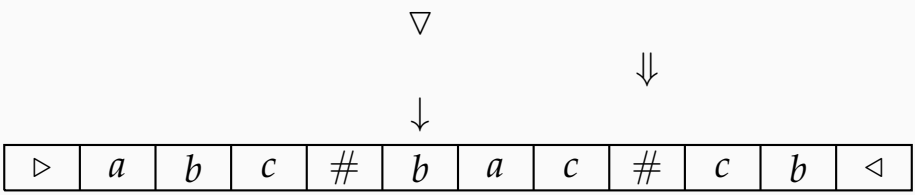
Output: $abcabc\#bac$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



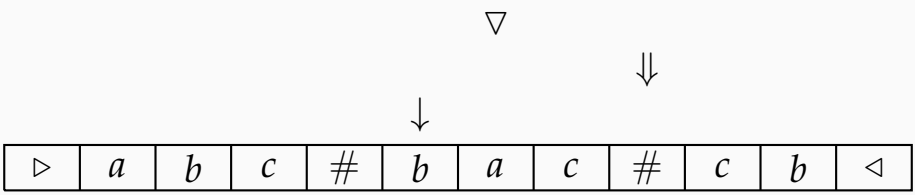
Output: *abcabc#bac*

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



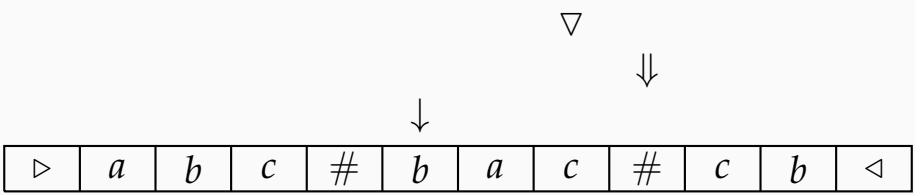
Output: *abcabc#bacb*

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



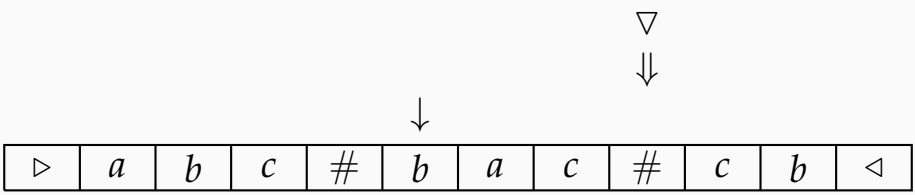
Output: *abcabc#bacba*

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



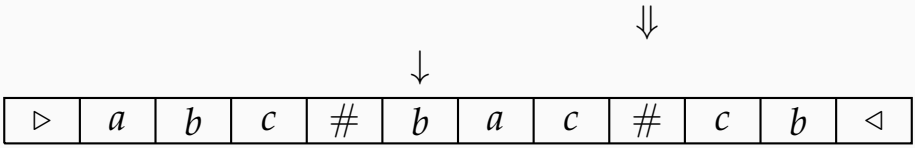
Output: $abcabc\#bacbac$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



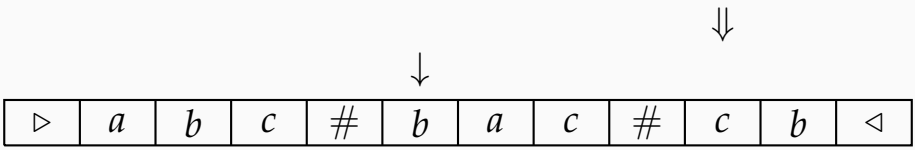
Output: *abcabc#bacbac*

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



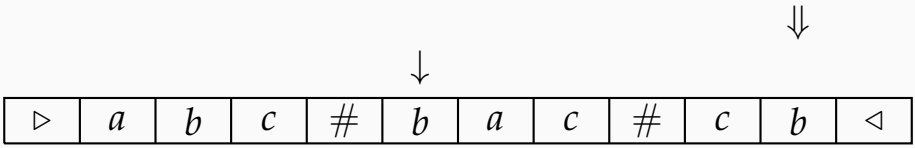
Output: *abcabc#bacbac*

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



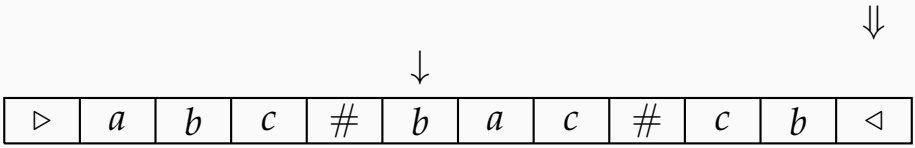
Output: *abcabc#bacbac*

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



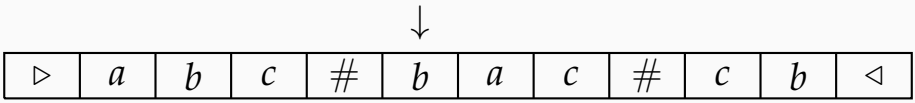
Output: $abcabc\#bacbac$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



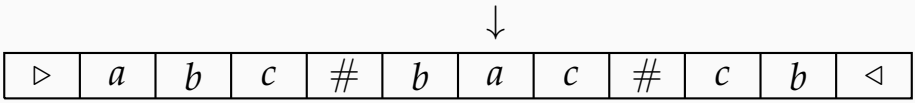
Output: $abcabc\#bacbac\#$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



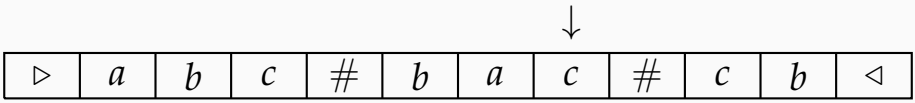
Output: $abcabc\#bacbac\#$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



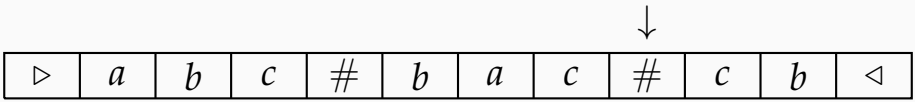
Output: $abcabc\#bacbac\#$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



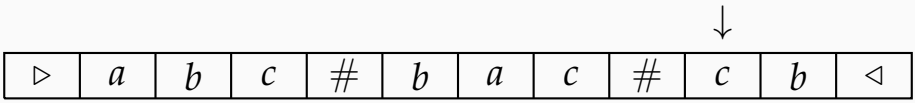
Output: $abcabc\#bacbac\#$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" `innsq`: $w_0\# \dots \# w_n \longmapsto (w_0)^n \# \dots \# (w_n)^n$



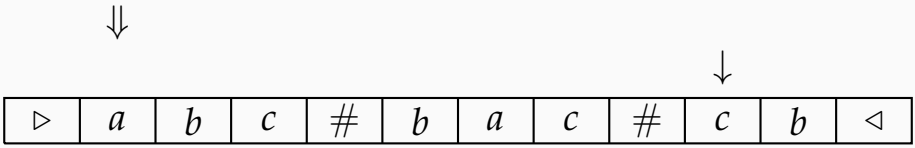
Output: *abcabc#bacbac#*

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



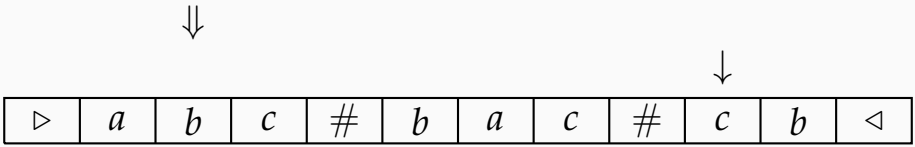
Output: $abcabc\#bacbac\#$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



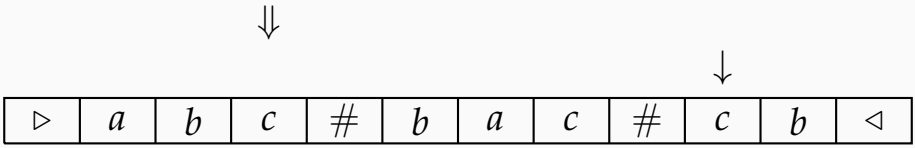
Output: $abcabc\#bacbac\#$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



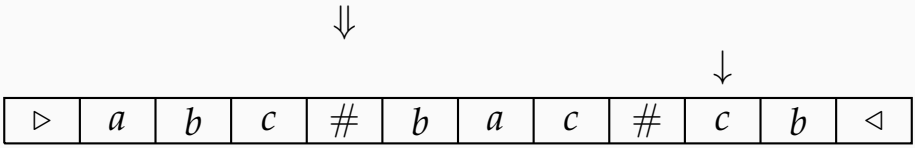
Output: $abcabc\#bacbac\#$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



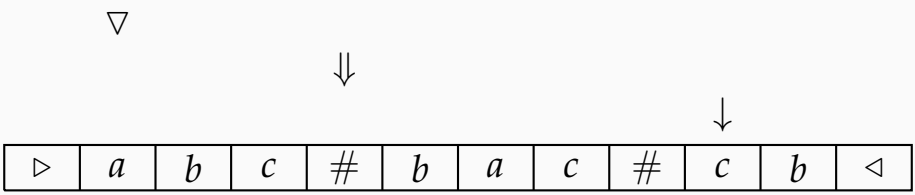
Output: $abcabc\#bacbac\#$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



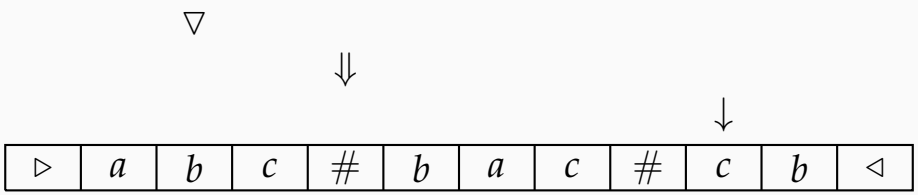
Output: *abcabc#bacbac#*

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



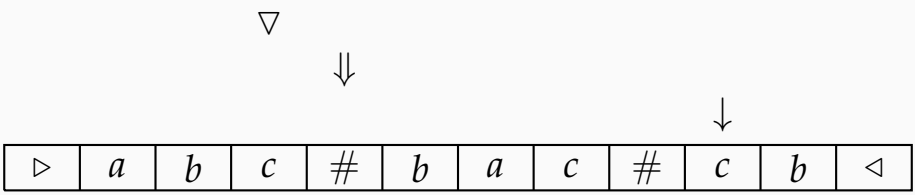
Output: *abcabc#bacbac#*

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



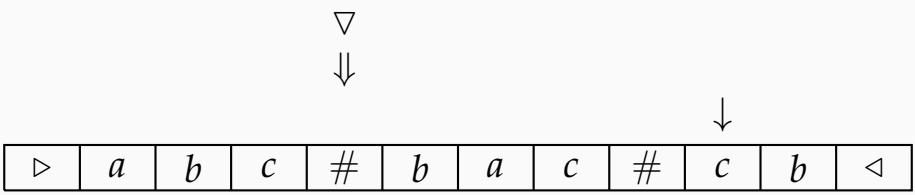
Output: *abcabc#bacbac#*

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



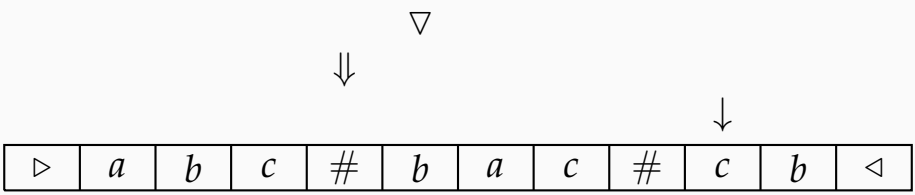
Output: *abcabc#bacbac#*

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



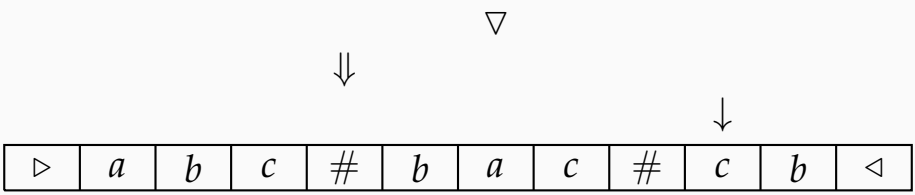
Output: $abcabc\#bacbac\#$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



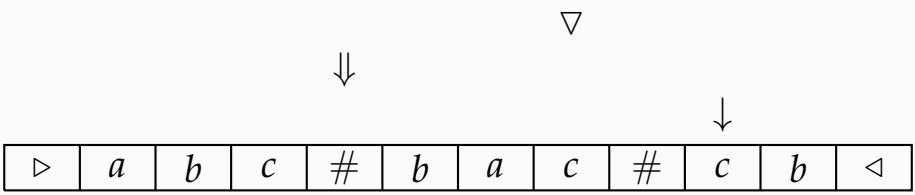
Output: $abcabc\#bacbac\#$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



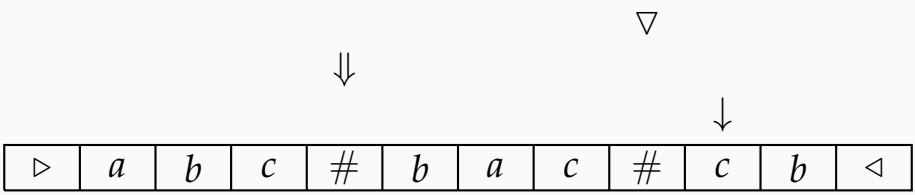
Output: *abcabc#bacbac#*

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



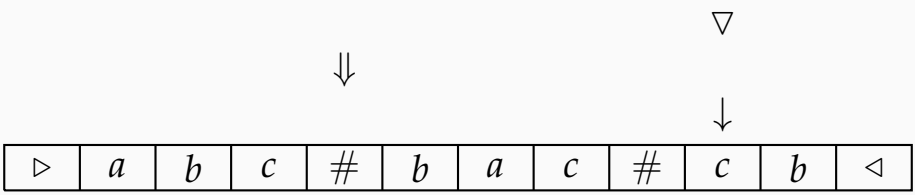
Output: $abcabc\#bacbac\#$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



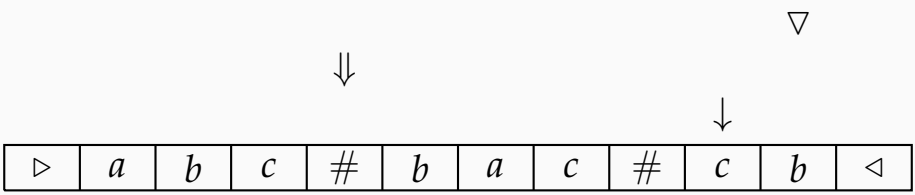
Output: $abcabc\#bacbac\#$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



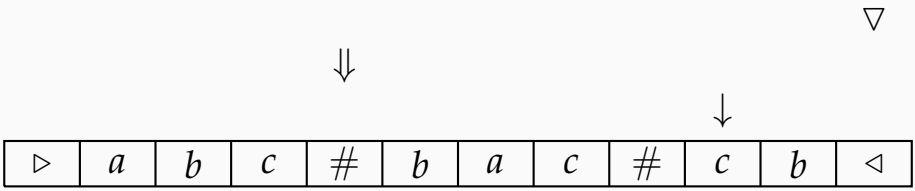
Output: $abcabc\#bacbac\#c$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



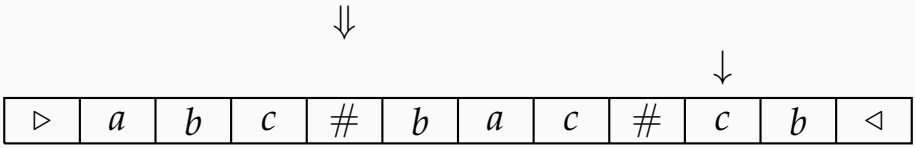
Output: $abcabc\#bacbac\#cb$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



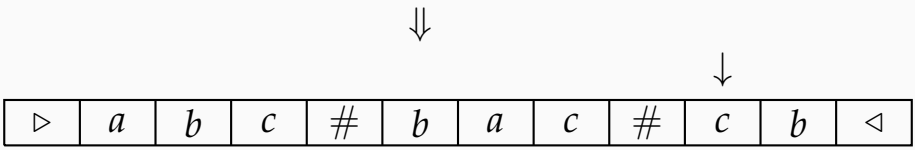
Output: $abcabc\#bacbac\#cb$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



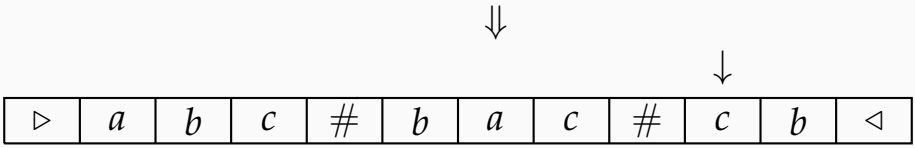
Output: $abcabc\#bacbac\#cb$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



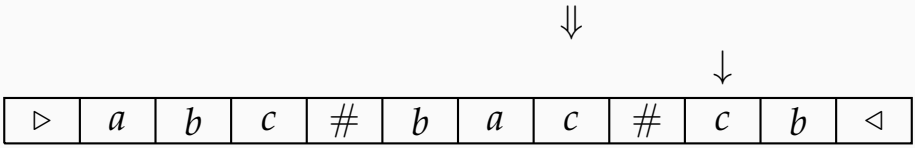
Output: $abcabc\#bacbac\#cb$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



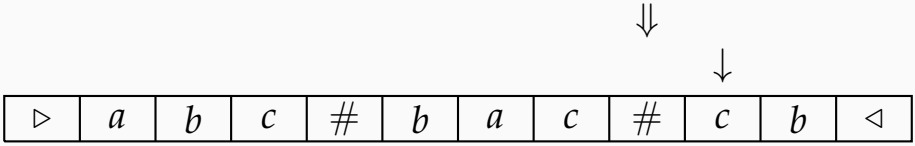
Output: $abcabc\#bacbac\#cb$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



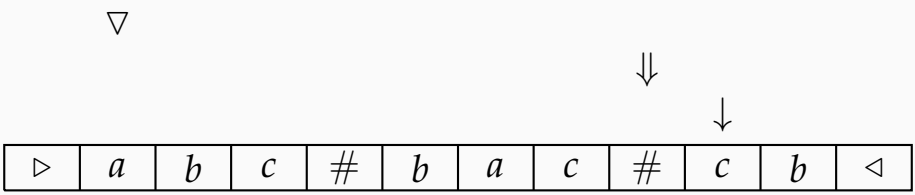
Output: $abcabc\#bacbac\#cb$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



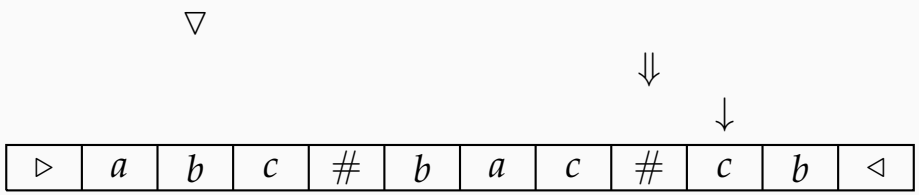
Output: $abcabc\#bacbac\#cb$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



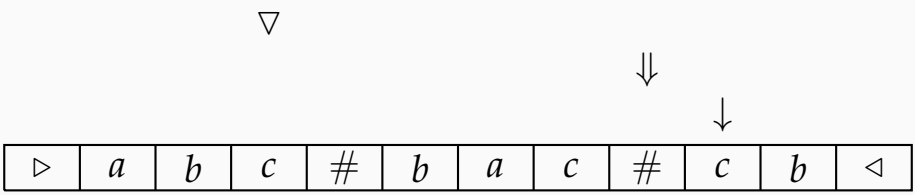
Output: *abcabc#bacbac#cb*

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



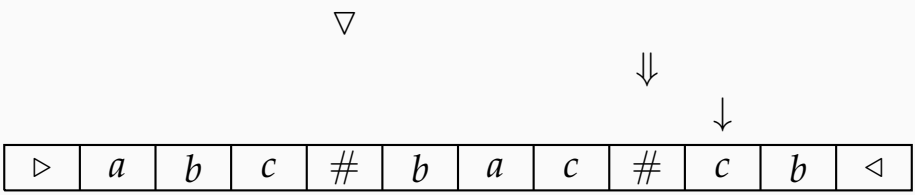
Output: *abcabc#bacbac#cb*

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



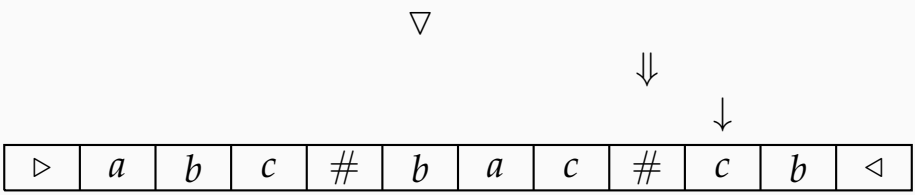
Output: *abcabc#bacbac#cb*

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



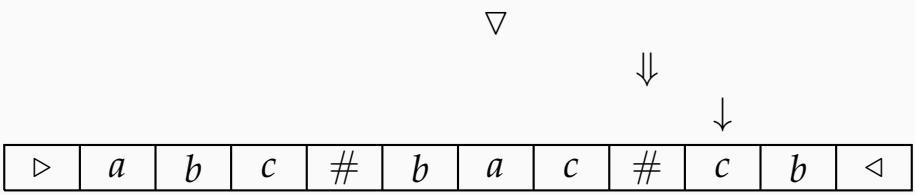
Output: *abcabc#bacbac#cb*

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



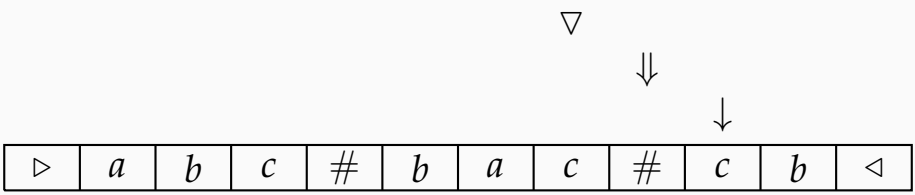
Output: $abcabc\#bacbac\#cb$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



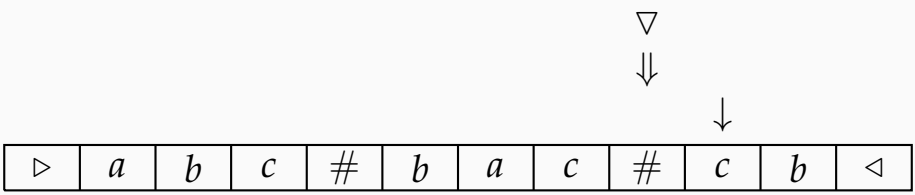
Output: $abcabc\#bacbac\#cb$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



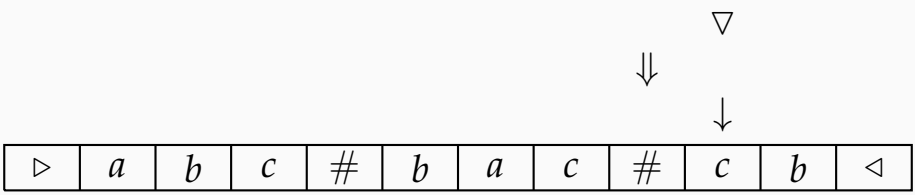
Output: $abcabc\#bacbac\#cb$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



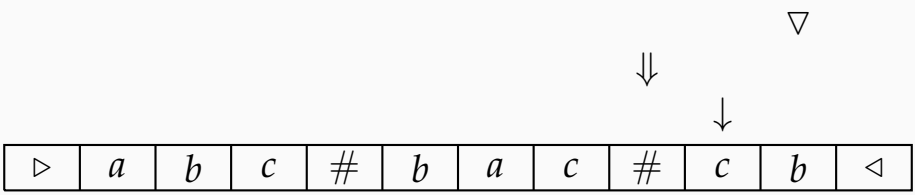
Output: $abcabc\#bacbac\#cb$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



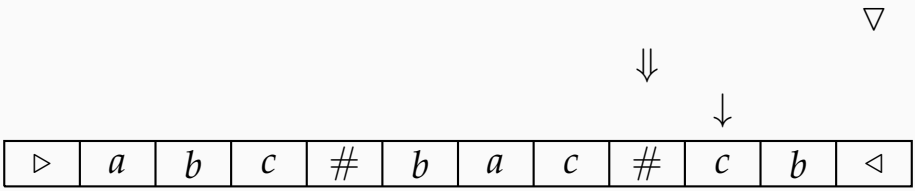
Output: $abcabc\#bacbac\#cbc$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



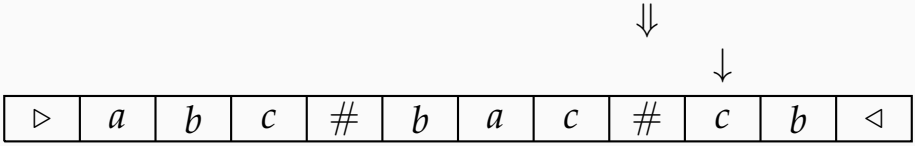
Output: $abcabc\#bacbac\#cbcb$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



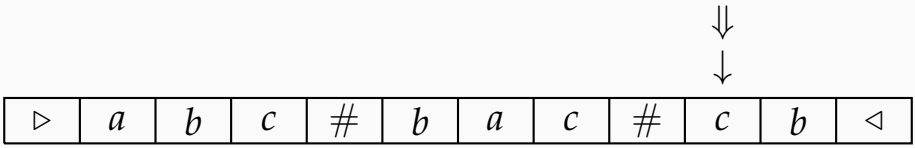
Output: $abcabc\#bacbac\#cbcb$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



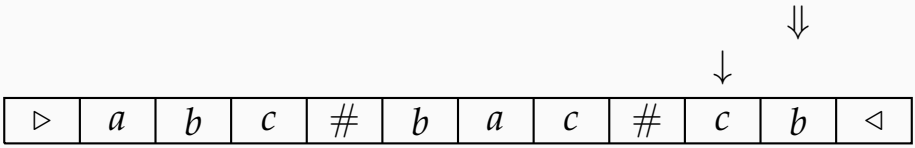
Output: *abcabc#bacbac#cbcb*

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



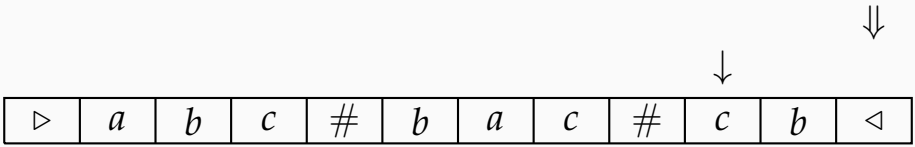
Output: $abcabc\#bacbac\#cbcb$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



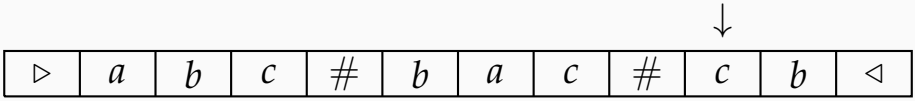
Output: $abcabc\#bacbac\#cbcb$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



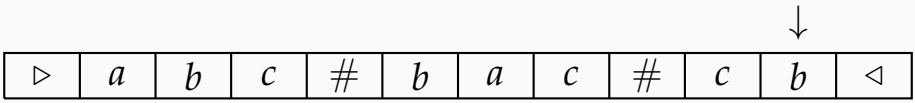
Output: $abcabc\#bacbac\#cbcb$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



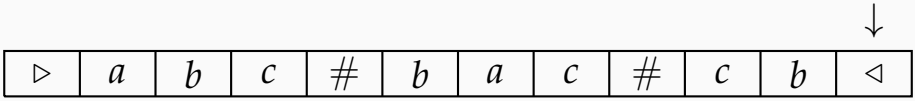
Output: $abcabc\#bacbac\#cbcb$

Pebble transducers [Milo, Suciu & Vianu 2000]

Polyregular functions = computed by k-pebble transducers ($k \geq 1$)

Finite states + *stack* of height $\leqslant k$ of two-way heads ("pebbles")

"Inner squaring" $\quad \texttt{innsq}\colon w_0\#\ldots\#w_n \longmapsto (w_0)^n\#\ldots\#(w_n)^n$



Output: $abcabc\#bacbac\#cbcb$

Polyregular functions and their growth

- Closed under composition [Engelfriet & Maneth 2002; Engelfriet 2015]
- Several alternative definitions in the last few years $\rightarrow$ revived interest
  [Bojańczyk 2018, 2023; Bojańczyk, Kiefer & Lhote 2019]
- Polynomial growth: k pebbles $\implies$ $O(n^k)$ growth

Polyregular functions and their growth

- Closed under composition [Engelfriet & Maneth 2002; Engelfriet 2015]
- Several alternative definitions in the last few years $\rightarrow$ revived interest
  [Bojańczyk 2018, 2023; Bojańczyk, Kiefer & Lhote 2019]
- Polynomial growth: k pebbles $\implies$ $O(n^k)$ growth

What about the converse?

For $w = w_0\# \dots \#w_n$, $|\texttt{innsq}(w)| = |(w_0)^n\# \dots \#(w_n)^n| = O(|w|^2)$

$\longrightarrow$ could `innsq` be computed with only 2 pebbles instead of 3?

Polyregular functions and their growth

- Closed under composition [Engelfriet & Maneth 2002; Engelfriet 2015]
- Several alternative definitions in the last few years $\rightarrow$ revived interest
  [Bojańczyk 2018, 2023; Bojańczyk, Kiefer & Lhote 2019]
- Polynomial growth: k pebbles $\implies$ $O(n^k)$ growth

What about the converse?

For $w = w_0\# \dots \#w_n$, $|\texttt{innsq}(w)| = |(w_0)^n\# \dots \#(w_n)^n| = O(|w|^2)$
$\longrightarrow$ could `innsq` be computed with only 2 pebbles instead of 3?

- Main theorem of a LICS'20 paper: $O(n^k)$ $\implies$ computable with k pebbles

Polyregular functions and their growth

- Closed under composition [Engelfriet & Maneth 2002; Engelfriet 2015]
- Several alternative definitions in the last few years $\rightarrow$ revived interest
  [Bojańczyk 2018, 2023; Bojańczyk, Kiefer & Lhote 2019]
- Polynomial growth: k pebbles $\implies$ $O(n^k)$ growth

What about the converse?

For $w = w_0\# \dots \#w_n$, $|\texttt{innsq}(w)| = |(w_0)^n\# \dots \#(w_n)^n| = O(|w|^2)$
$\longrightarrow$ could `innsq` be computed with only 2 pebbles instead of 3?

- Main theorem of a LICS'20 paper: $O(n^k)$ $\implies$ computable with k pebbles
- But actually, `innsq` requires 3 pebbles! [Bojańczyk 2023]

Pebble non-minimization

More than that, we have:

For any $k \geq 2$ there is a polyregular f with $|f(w)| = O(|w|^2)$ that needs k pebbles.

$\longrightarrow$ the most technical proof in Bojańczyk's LICS'23 paper...

Pebble non-minimization

More than that, we have:

For any $k \geq 2$ there is a polyregular f with $|f(w)| = O(|w|^2)$ that needs k pebbles.

$\longrightarrow$ the most technical proof in Bojańczyk's LICS'23 paper... our easier alternative:

Theorem (Kiefer, N. & Pradic, arXiv preprint 2023)

For any $k \geq 1$ there is a polyregular f with $|f(w)| = O(|w|^2)$ whose output language differs from that of any function at level k of "Engelfriet's class".

- k-pebble $\subset$ level-k [Engelfriet & Maneth 2003] $\rightarrow$ recover previous result
- this appears without the "$|f(w)| = O(|w|^2)$" in [EM02]; we adapt the proof

Proof strategy for pebble non-minimization (1)

Theorem (Engelfriet & Maneth 2002)

If $L' \subseteq (\Sigma \cup \Delta)^*$ is d-complete (cf. below) for $L \subseteq \Sigma^*$ and L' is the output language of a level-$(k+1)$ function, then L is the output language of a level-k function.

L' is *d-complete for* L when for every $u \in L$ there exist $w_0, \ldots, w_n \in \Delta^*$ such that

- all the w_i for $i \in \{1, \ldots, n-1\}$ are *pairwise distinct* words;
- $n = |u|$ and $w_0 u[1] w_1 \ldots u[n] w_n \in L'$;

+ "conversely", by erasing the letters from Δ in the words in L' one gets exactly L

Idea

We build a sequence of functions such that $\mathrm{Im}(f_{k+1})$ is d-complete for $\mathrm{Im}(f_k)$, and $\mathrm{Im}(f_1)$ is *not* the output language of a level-1 function.

Proof strategy for pebble non-minimization (2)

We keep track of some kind of "origin semantics" in the inductive construction

- $f_1(a^n) = (a^{n-1}b)^{n-1}$, from $f_1^{(\mathbb{A})}\begin{pmatrix} a & a & a \\ 3 & 1 & 2 \end{pmatrix} = \begin{matrix} a & a & b & a & a & b \\ 3,3 & 3,1 & 3,2 & 1,3 & 1,1 & 1,2 \end{matrix}$

  $\rightsquigarrow f_2(a \bullet\bullet\bullet a \bullet a \bullet\bullet) = a\Box\Box\Box\Diamond\Diamond\Diamond a\Box\Box\Box\Diamond b\Box\Box\Box\Diamond\Diamond a\Box\Diamond\Diamond\Diamond a\Box\Diamond b\Box\Diamond\Diamond$

- distinct input factors in $\bullet^*$ $\implies$ distinct output factors in $\Box^*\Diamond^*$
  $\rightsquigarrow \mathrm{Im}(f_2)$ is d-complete for $\mathrm{Im}(f_1)$
- f_2 is induced by a suitable $f_2^{(\mathbb{A})}$, from which we define f_3, etc.

Positive results

Theorem (Bojańczyk LICS'23 / Kiefer, N. & Pradic, unpublished)

For any $k \geq 2$ there is a polyregular f with $|f(w)| = O(|w|^2)$ that needs k pebbles.

Is 2 the minimal exponent for this phenomenon?

Positive results

Theorem (Bojańczyk LICS'23 / Kiefer, N. & Pradic, unpublished)

For any $k \geq 2$ there is a polyregular f with $|f(w)| = O(|w|^2)$ that needs k pebbles.

Is 2 the minimal exponent for this phenomenon?

Theorem (Engelfriet, Inaba & Maneth 2021)

If a tree-to-tree function has linear growth and is in "Engelfriet's class", then it is definable by an MSO tree transduction.

This class contains polyregular functions (k-pebble $\subset$ level-k), so:

$$\text{polyregular} \cap O(n) \iff \text{MSOT} \iff \text{two-way transducer} = \text{1-pebble}$$

Positive results

Theorem (Bojańczyk LICS'23 / Kiefer, N. & Pradic, unpublished)

For any $k \geq 2$ there is a polyregular f with $|f(w)| = O(|w|^2)$ that needs k pebbles.

Is 2 the minimal exponent for this phenomenon?

Theorem (Engelfriet, Inaba & Maneth 2021)

If a tree-to-tree function has linear growth and is in "Engelfriet's class", then it is definable by an MSO tree transduction.

This class contains polyregular functions (k-pebble $\subset$ level-k), so:

$$\text{polyregular} \cap O(n) \iff \text{MSOT} \iff \text{two-way transducer} = \text{1-pebble}$$

Note: k-dimensional *MSO interpretations* $\iff$ polyregular $\cap\, O(n^k)$, for all k

Conclusion

A large variety of transducer models, but some "canonical" function classes

1. Regular functions (linear growth) e.g. `mapReverse`

   two-way transducers, copyless streaming string transducers, ...

2. Polyregular functions (polynomial growth) e.g. "inner squaring"

   pebble transducers, MSO interpretations, ...

3. "Engelfriet's class" (hyperexponential growth)

   compositions of macro tree transducers, iterated pushdown transducers, ...

Conclusion

A large variety of transducer models, but some "canonical" function classes

1. Regular functions (linear growth) e.g. `mapReverse`
   two-way transducers, copyless streaming string transducers, ...
2. Polyregular functions (polynomial growth) e.g. "inner squaring"
   pebble transducers, MSO interpretations, ...
3. "Engelfriet's class" (hyperexponential growth)
   compositions of macro tree transducers, iterated pushdown transducers, ...

(1) and (3) have a rich history.

(2) is newer, but we can apply results and tools from (3)!

Conclusion

A large variety of transducer models, but some "canonical" function classes

1. Regular functions (linear growth) e.g. `mapReverse`
   two-way transducers, copyless streaming string transducers, ...
2. Polyregular functions (polynomial growth) e.g. "inner squaring"
   pebble transducers, MSO interpretations, ...
3. "Engelfriet's class" (hyperexponential growth)
   compositions of macro tree transducers, iterated pushdown transducers, ...

(1) and (3) have a rich history.

(2) is newer, but we can apply results and tools from (3)!

Thank you for your attention!