Single use restriction

A new trend in transformations with atoms

Rafał Stefański

Joint work with Mikołaj Bojańczyk

Work in progress

Deterministic register automata

Fix a countably infinite set of *atoms*.
They can only be compared for equality.

$$\mathbb{A} = \{ 1\ 2\ 3 \ldots \}$$

model by Kaminski, Francez 1994

Deterministic register automata

The first letter appears again

1 3 2 4 2 1 5

Deterministic register automata

The first letter appears again

$q_{start}(\bot)$

↓

1 3 2 4 2 1 5

Deterministic register automata

The first letter appears again

$q_{scan}(\quad)$

↓

1 3 2 4 2 1 5

Deterministic register automata

The first letter appears again

$q_{scan}(1)$

↓

1 3 2 4 2 1 5

Deterministic register automata

The first letter appears again

$q_{scan}(1)$

↓

1 3 2 4 2 1 5

Deterministic register automata

The first letter appears again

$q_{scan}(1)$

↓

1 3 2 4 2 1 5

Deterministic register automata

The first letter appears again

$q_{scan}(1)$

↓

1 3 2 4 2 1 5

Deterministic register automata

The first letter appears again

$q_{scan}(1)$

1 3 2 4 2 1 5

Deterministic register automata

The first letter appears again

Deterministic register automata

The first letter appears again

$q_{found}(\bot)$

↓

1 3 2 4 2 1 5

Deterministic register automata

The first letter appears again

1 3 2 4 2 1 5

Single use restriction

Every read access destroys the contents of the register.

Single use restriction

Every read access destroys the contents of the register.

$q_{start}(\bot)$

↓

1 3 2 4 2 1 5

Single use restriction

Every read access destroys the contents of the register.

$q_{scan}(1)$

↓

1 3 2 4 2 1 5

Single use restriction

Every read access destroys the contents of the register.

$$q_{scan}(\bot)$$

$$\downarrow$$

1 3 2 4 2 1 5

Single use restriction

Every read access destroys the contents of the register.

???

$q_{scan}(\perp)$

↓

1 3 2 4 2 1 5

Single use restriction

Every read access destroys the contents of the register.

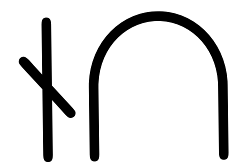
???

$q_{scan}(\perp)$

↓

1 3 2 4 2 1 5 ?

Single use register automata

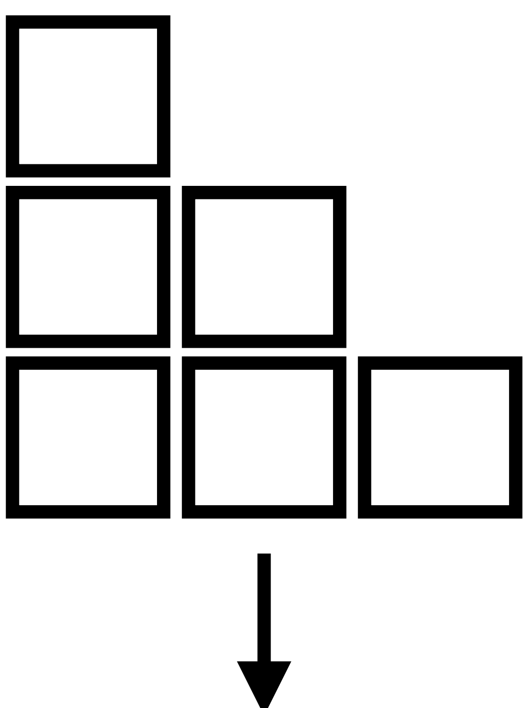


The first letter appears again

Register automata

Single use register automata

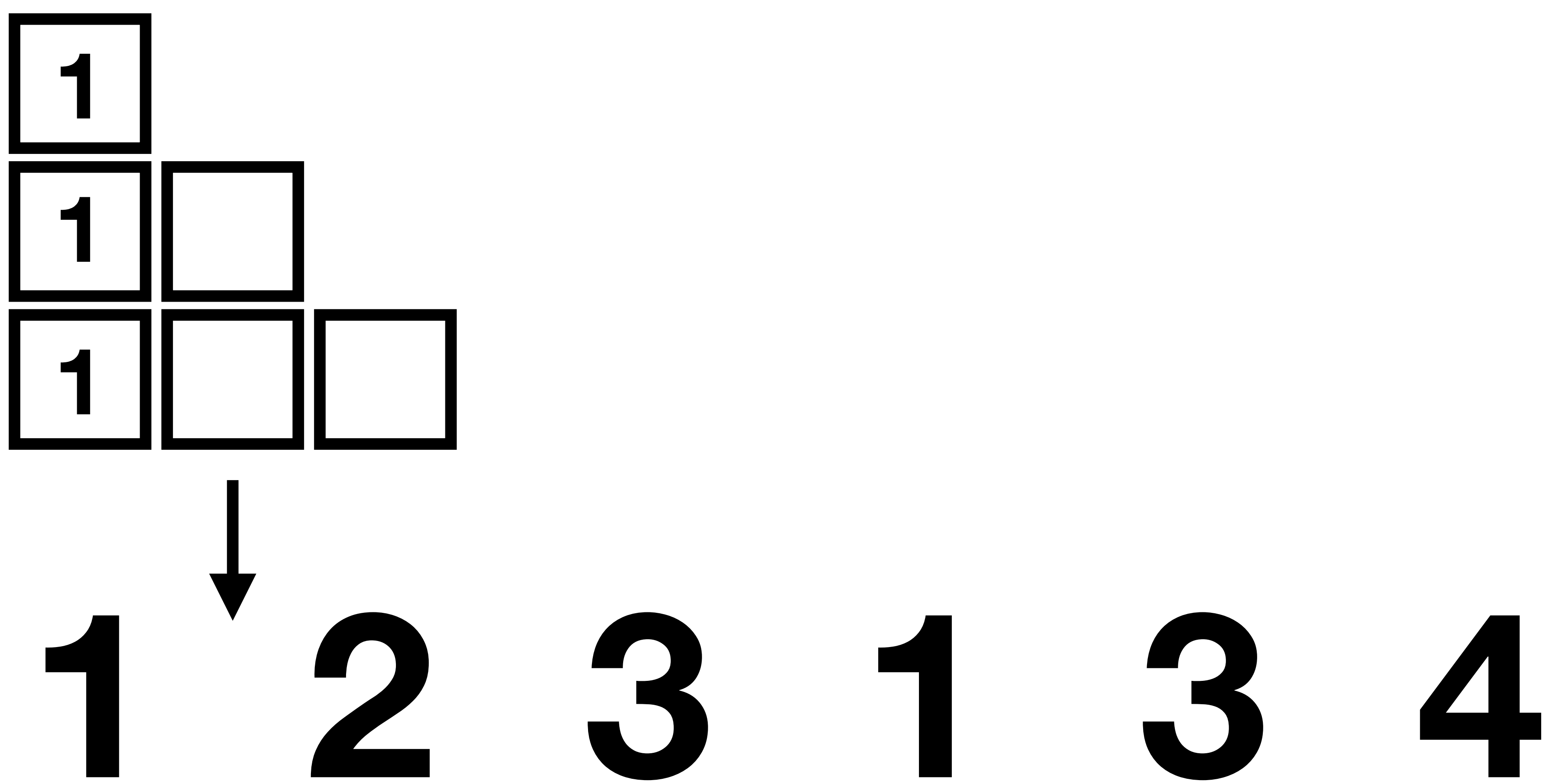
There are at most 3 distinct letters



1 2 3 1 3 4

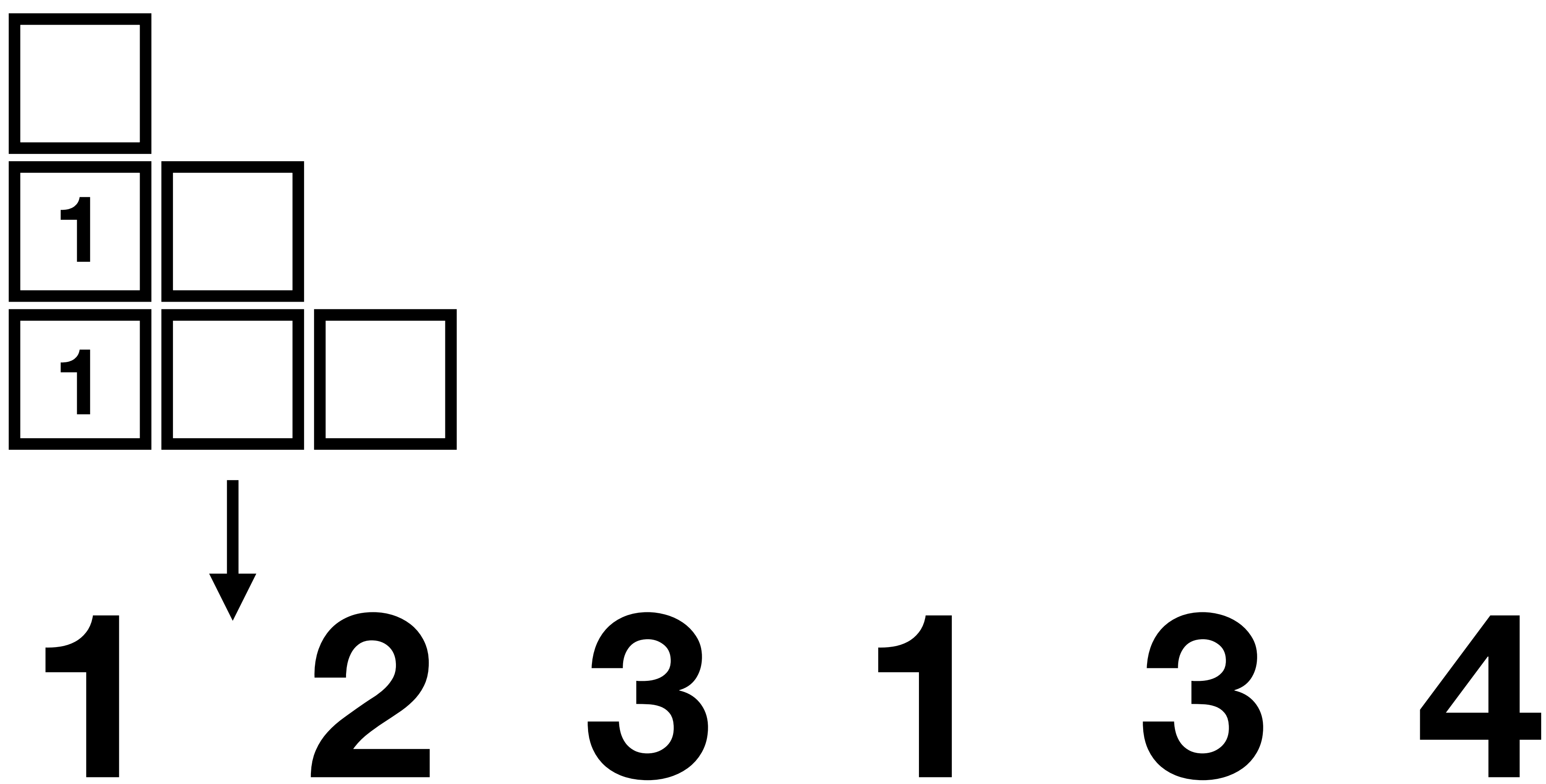
Single use register automata

There are at most 3 distinct letters



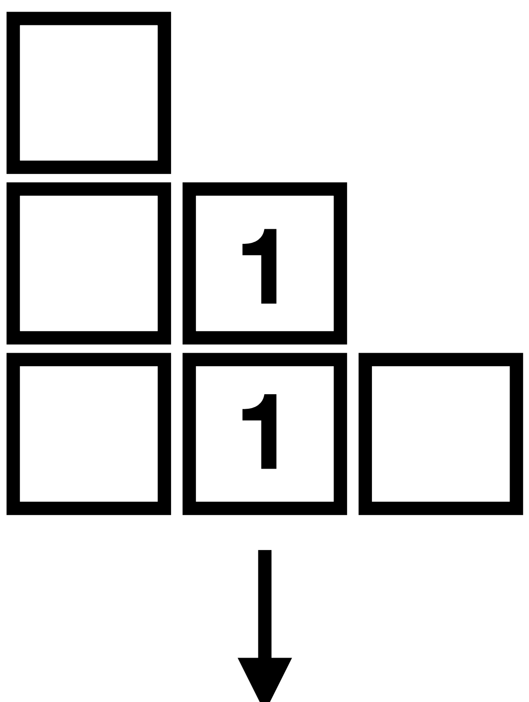
Single use register automata

There are at most 3 distinct letters



Single use register automata

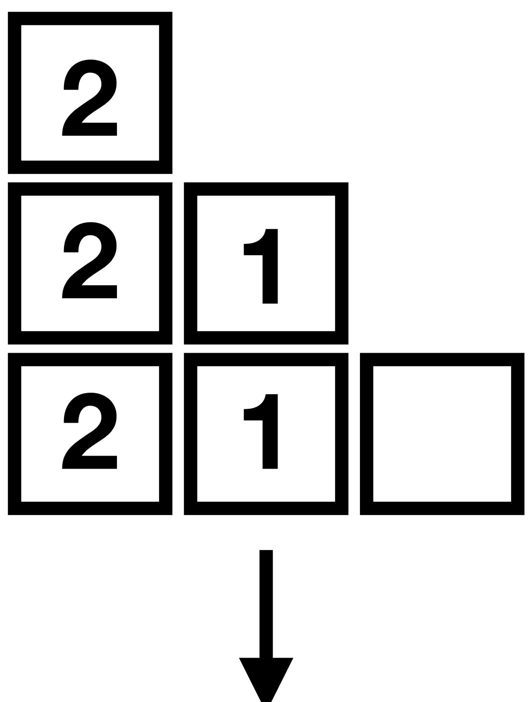
There are at most 3 distinct letters



1 2 3 1 3 4

Single use register automata

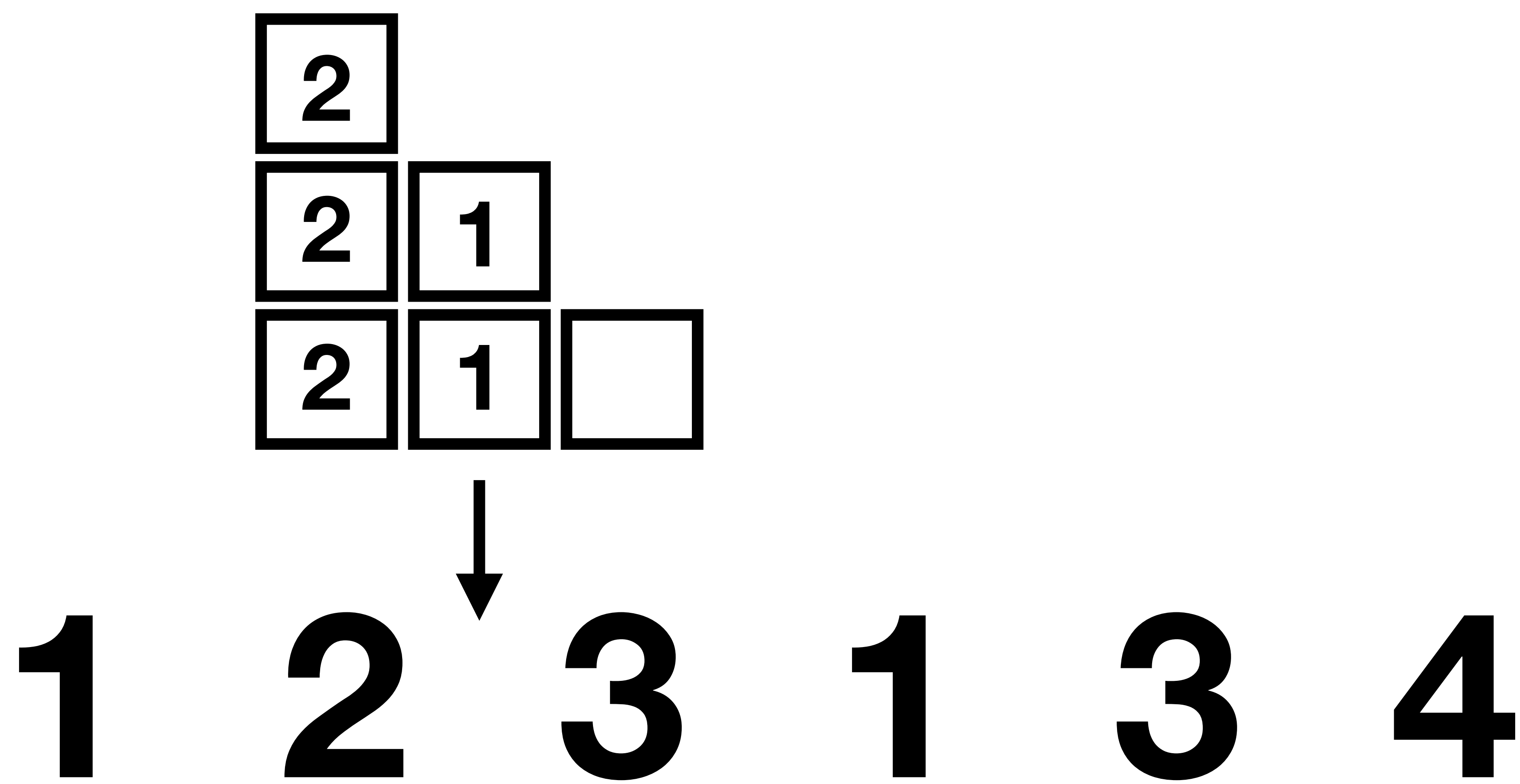
There are at most 3 distinct letters



1 2 3 1 3 4

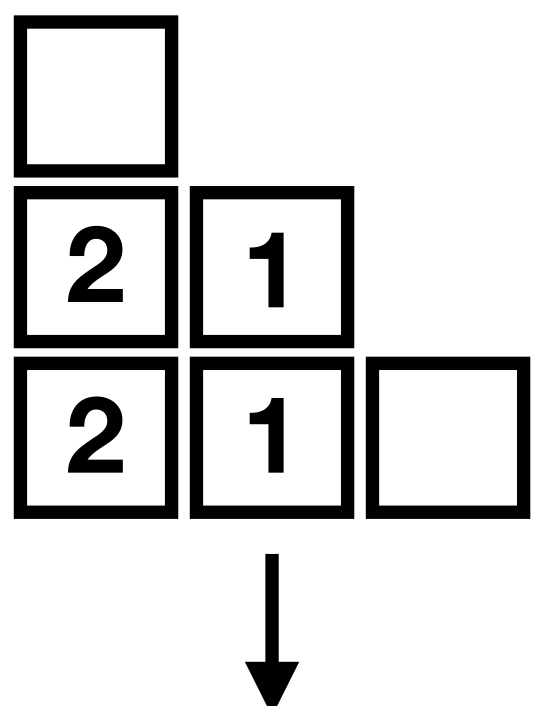
Single use register automata

There are at most 3 distinct letters



Single use register automata

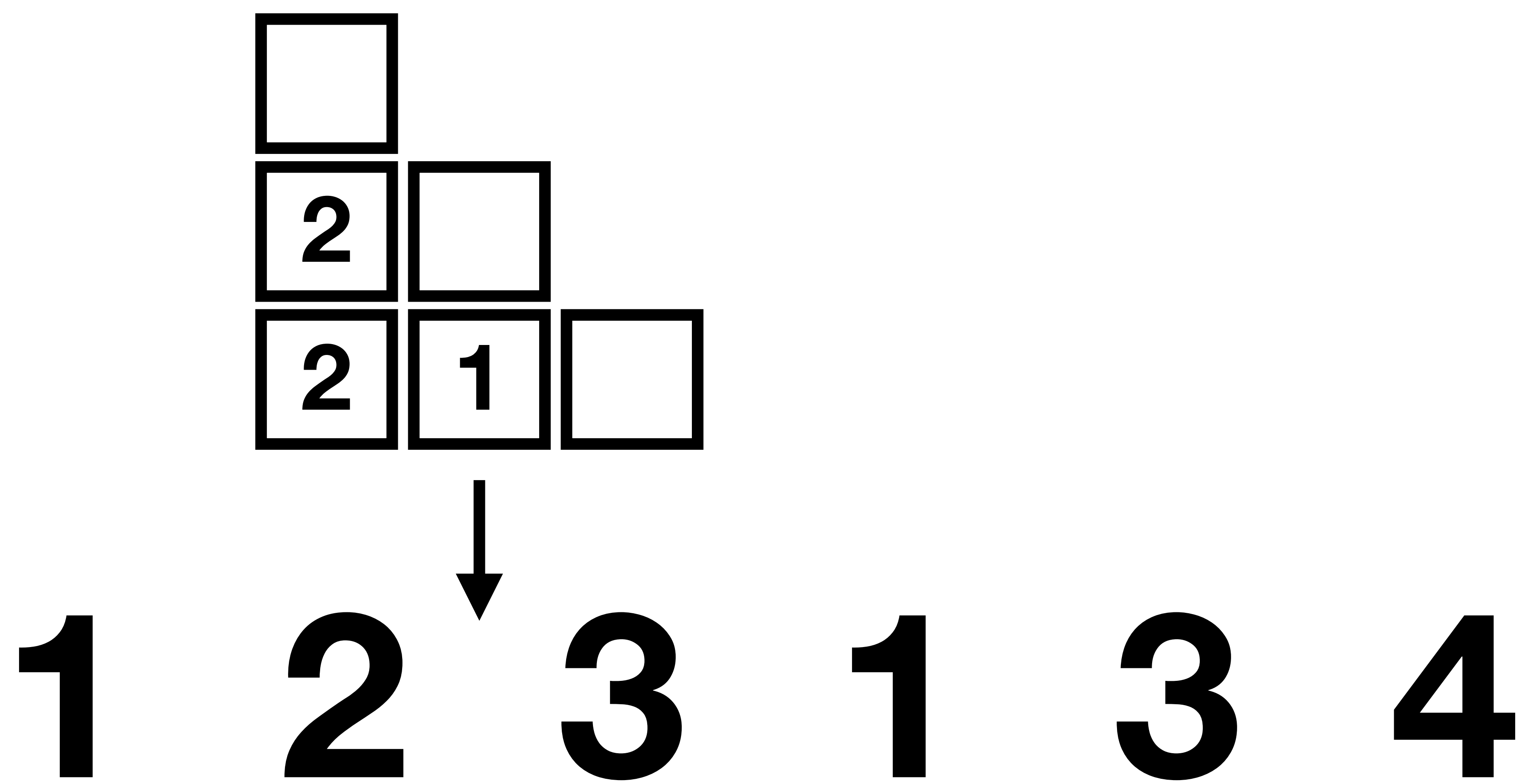
There are at most 3 distinct letters



1 2 3 1 3 4

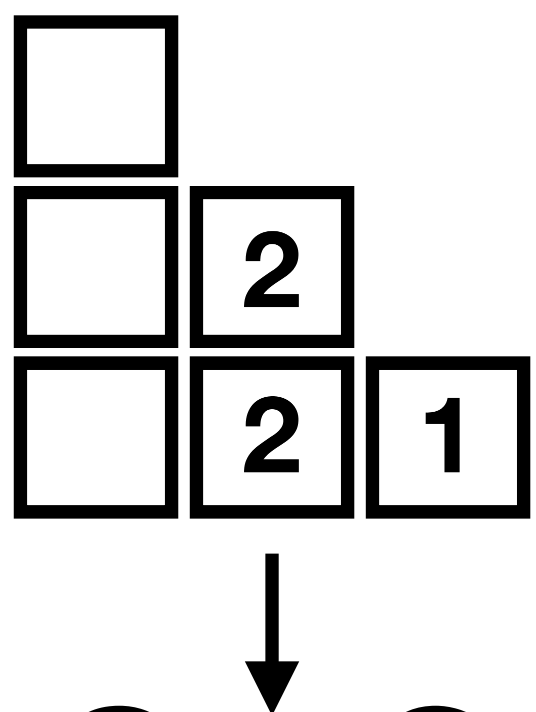
Single use register automata

There are at most 3 distinct letters



Single use register automata

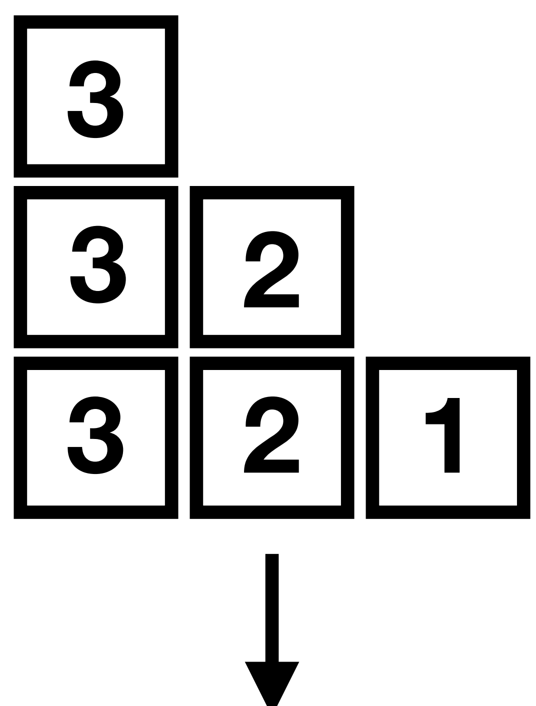
There are at most 3 distinct letters



1 2 3 1 3 4

Single use register automata

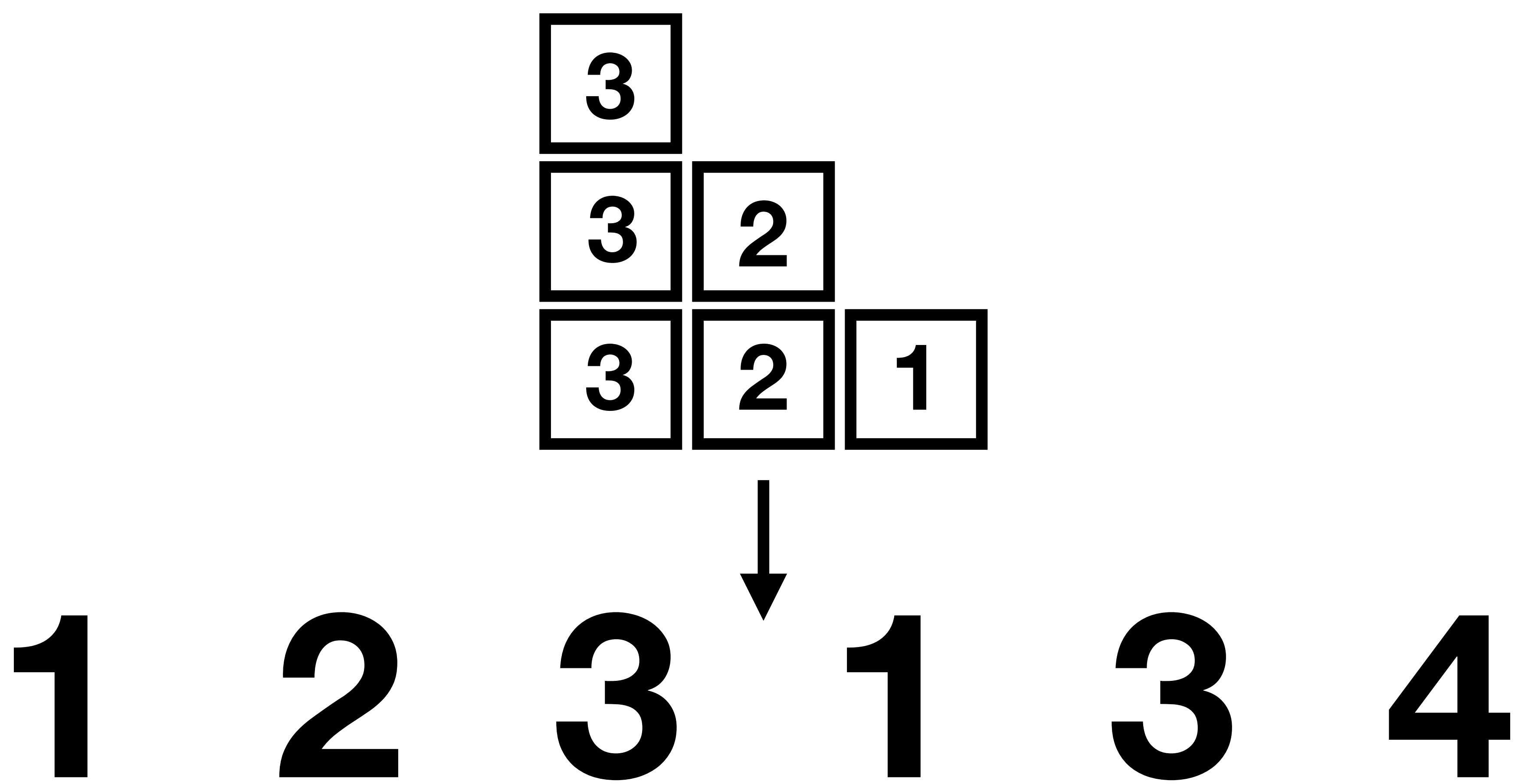
There are at most 3 distinct letters



1 2 3 1 3 4

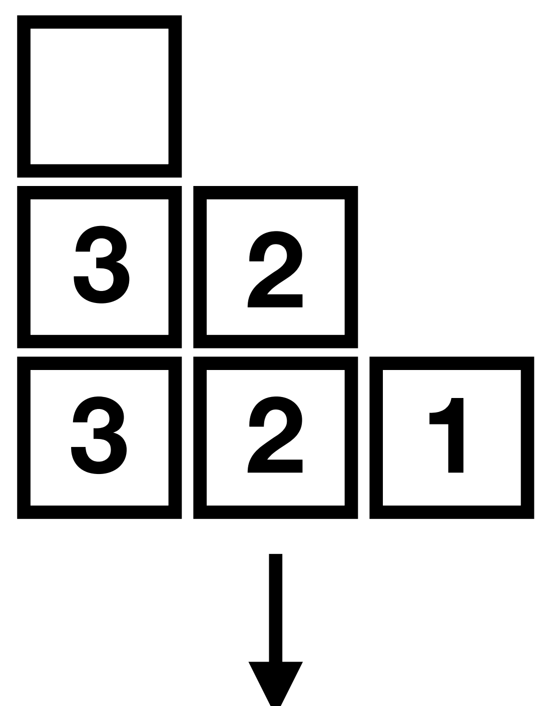
Single use register automata

There are at most 3 distinct letters



Single use register automata

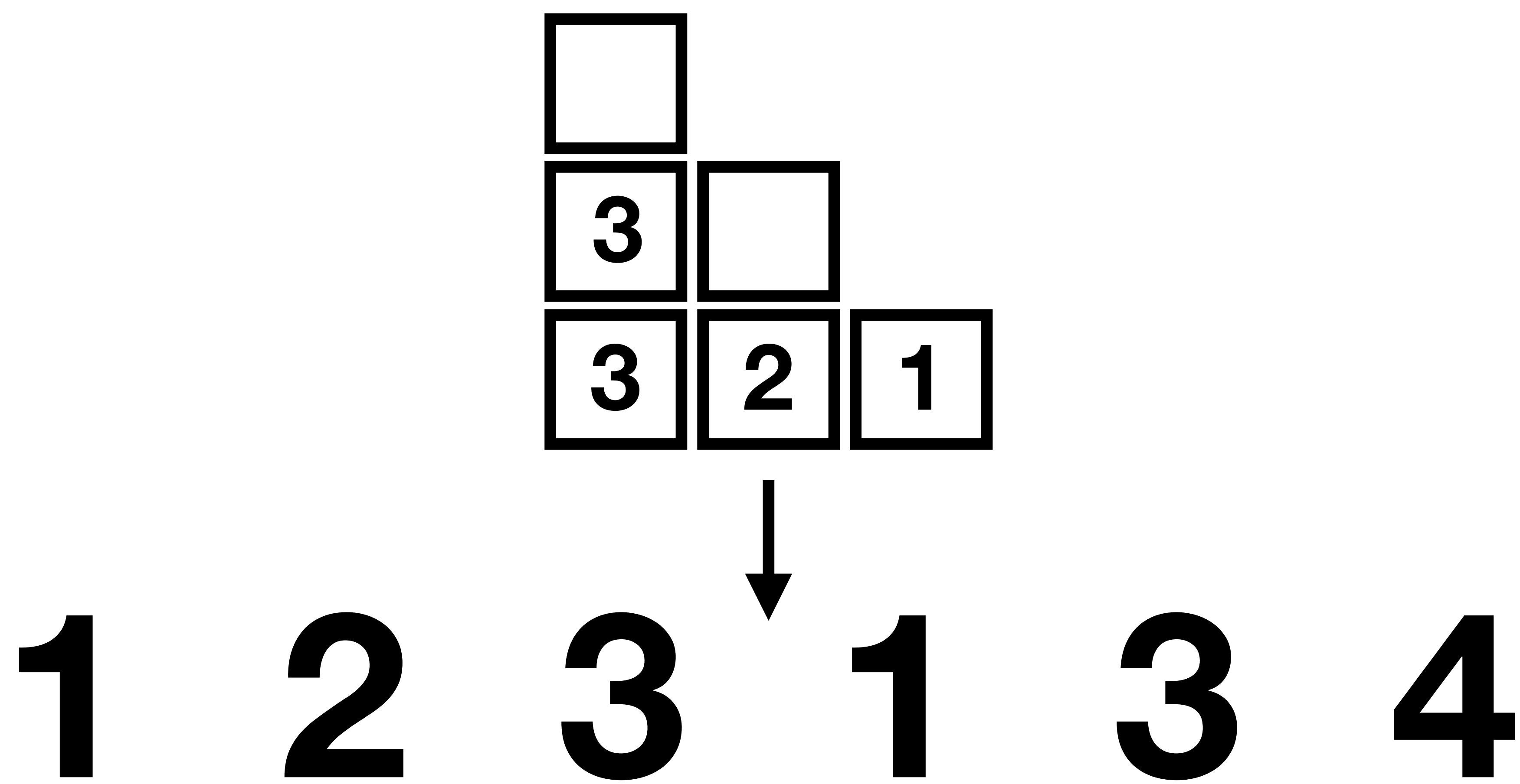
There are at most 3 distinct letters



1 2 3 1 3 4

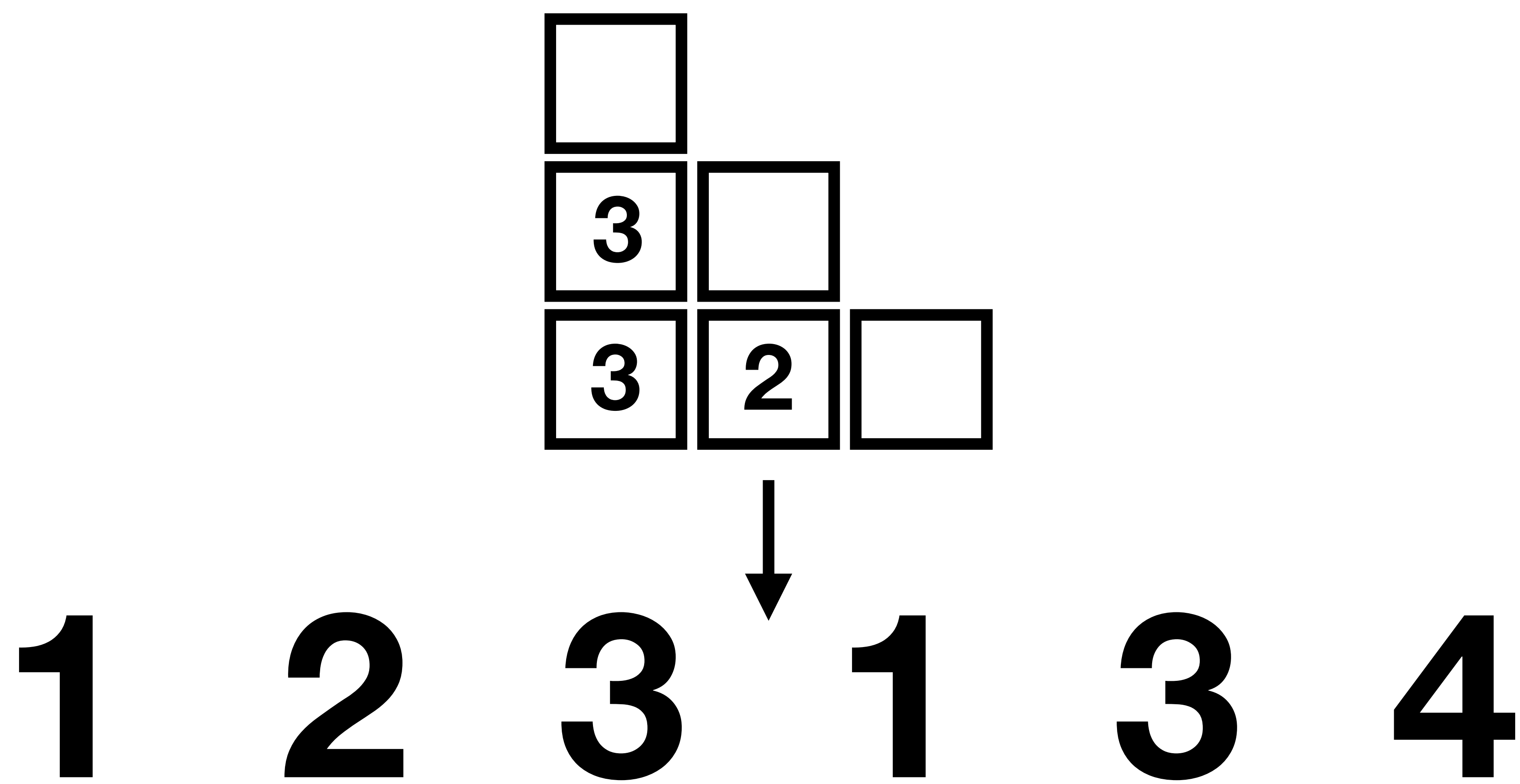
Single use register automata

There are at most 3 distinct letters



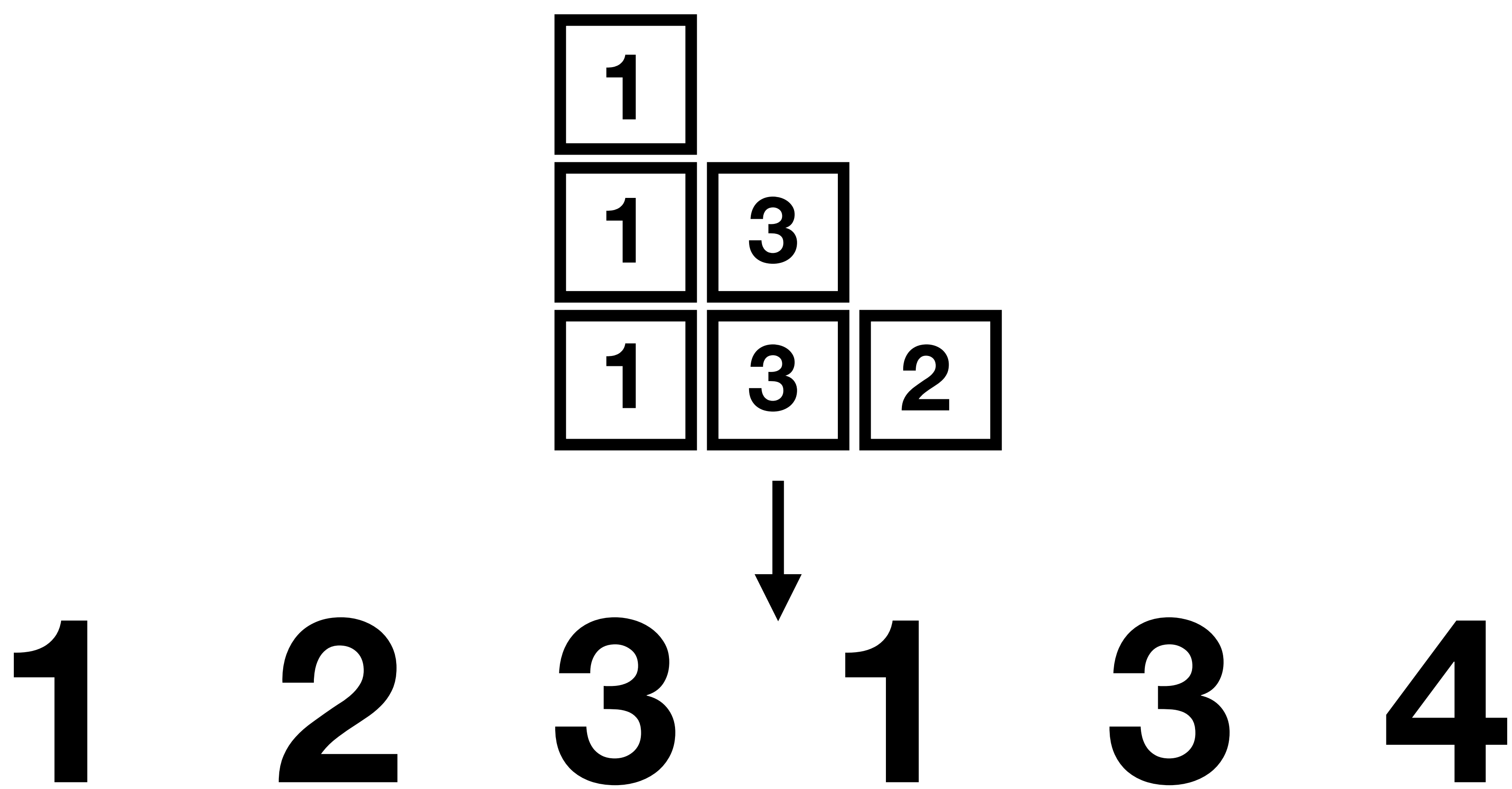
Single use register automata

There are at most 3 distinct letters



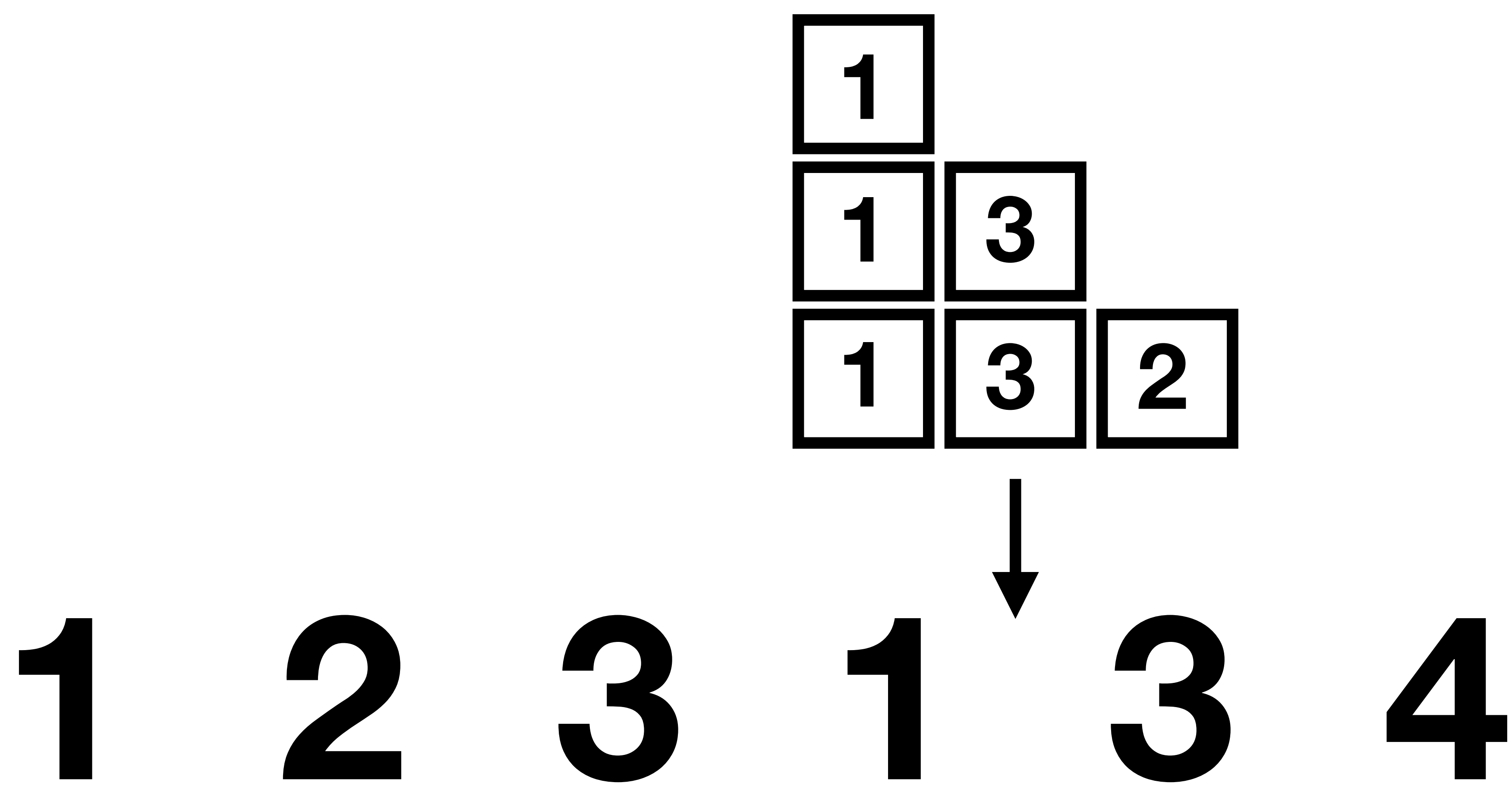
Single use register automata

There are at most 3 distinct letters



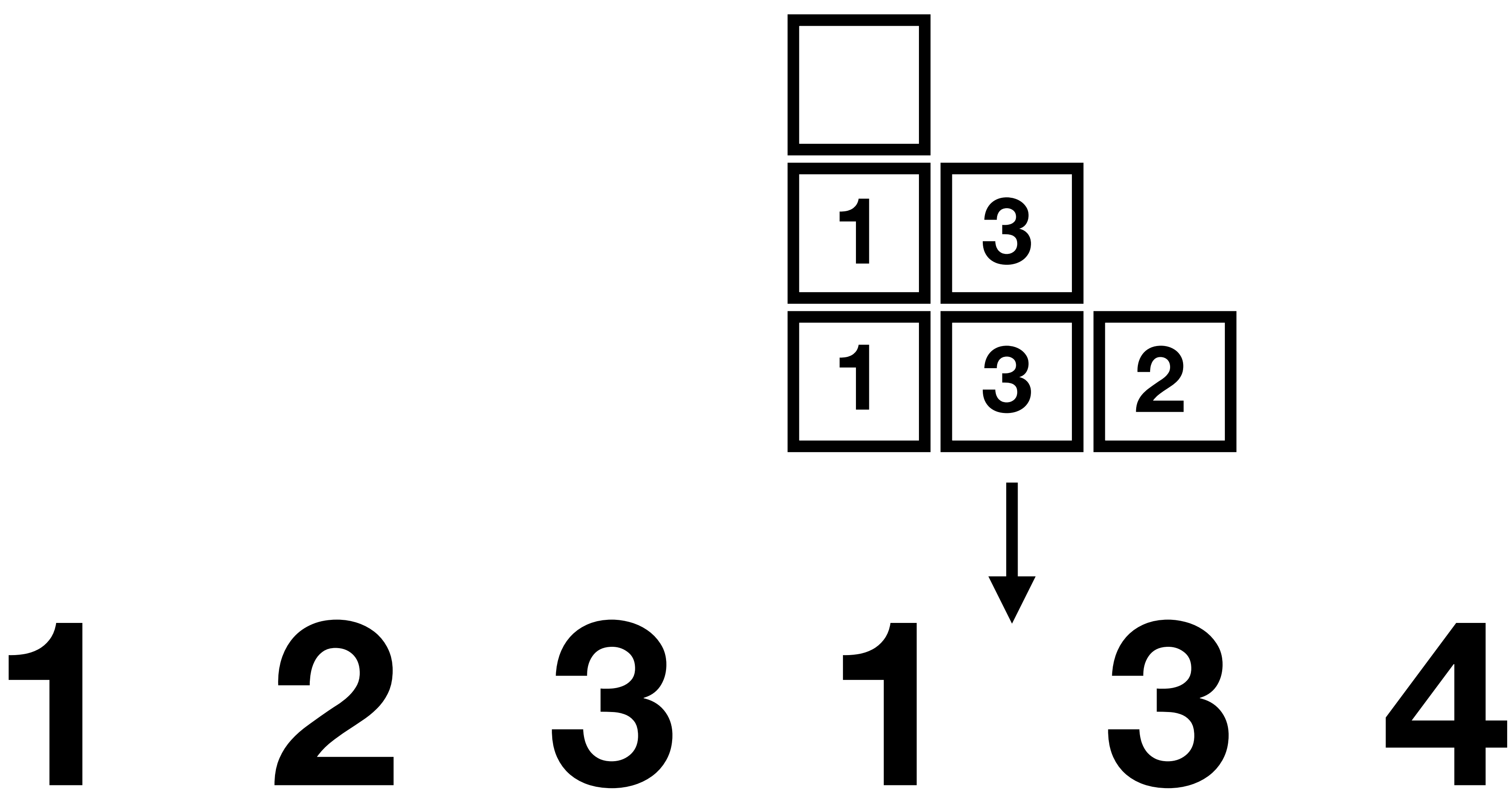
Single use register automata

There are at most 3 distinct letters



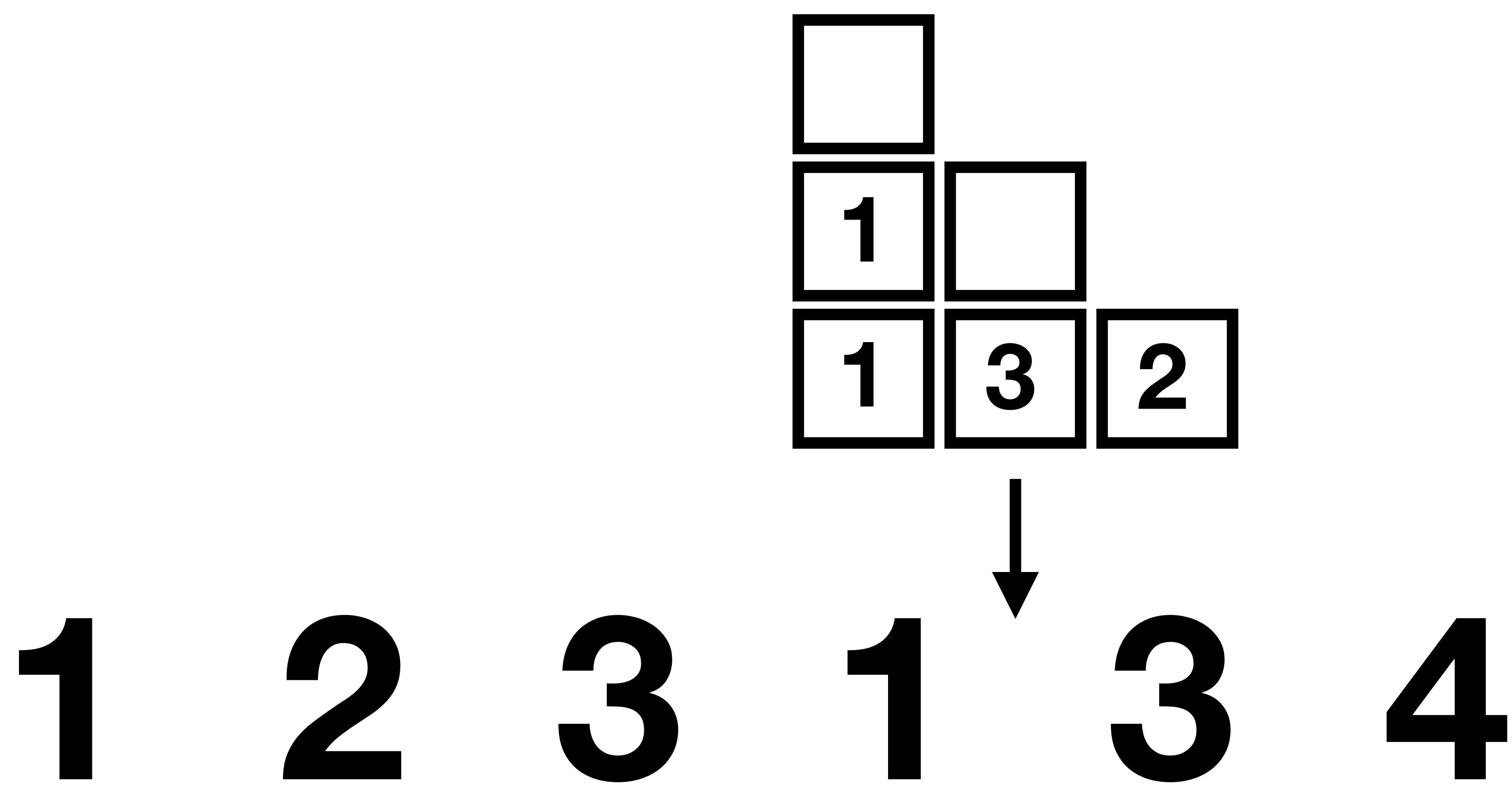
Single use register automata

There are at most 3 distinct letters



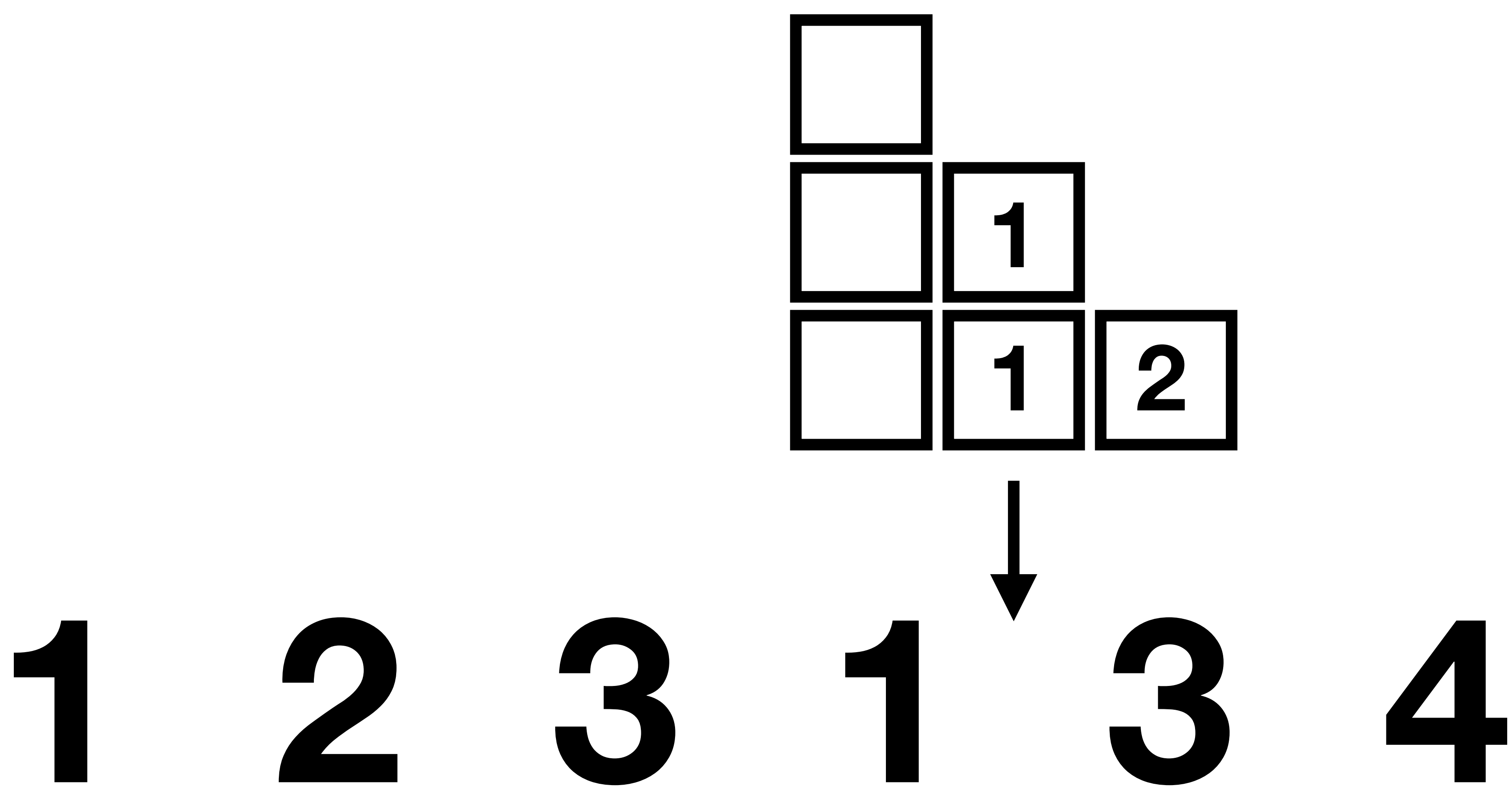
Single use register automata

There are at most 3 distinct letters



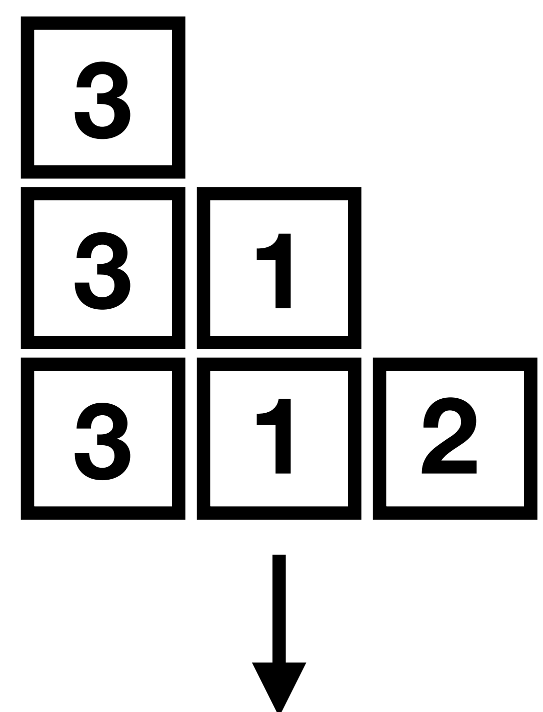
Single use register automata

There are at most 3 distinct letters



Single use register automata

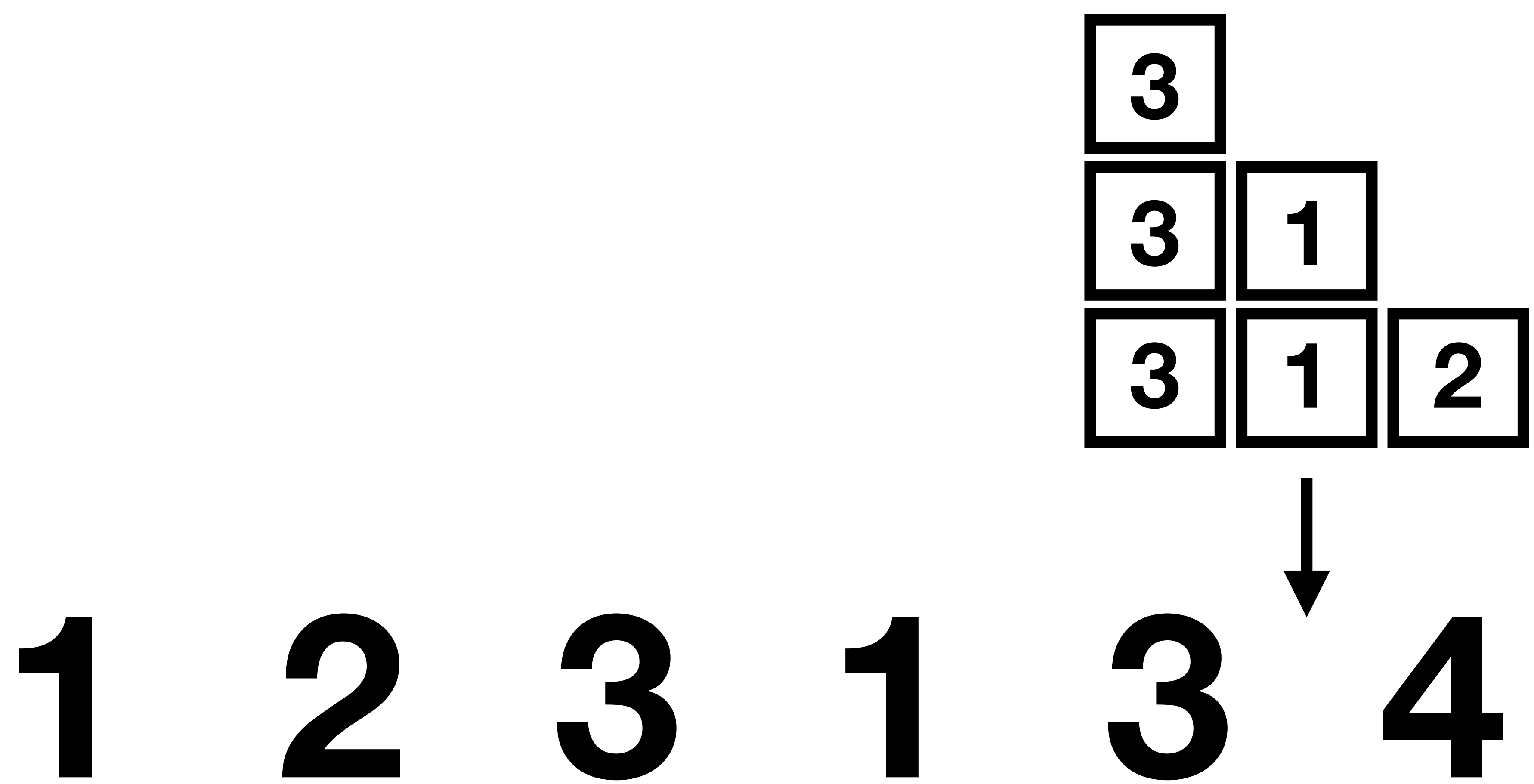
There are at most 3 distinct letters



1 2 3 1 3 4

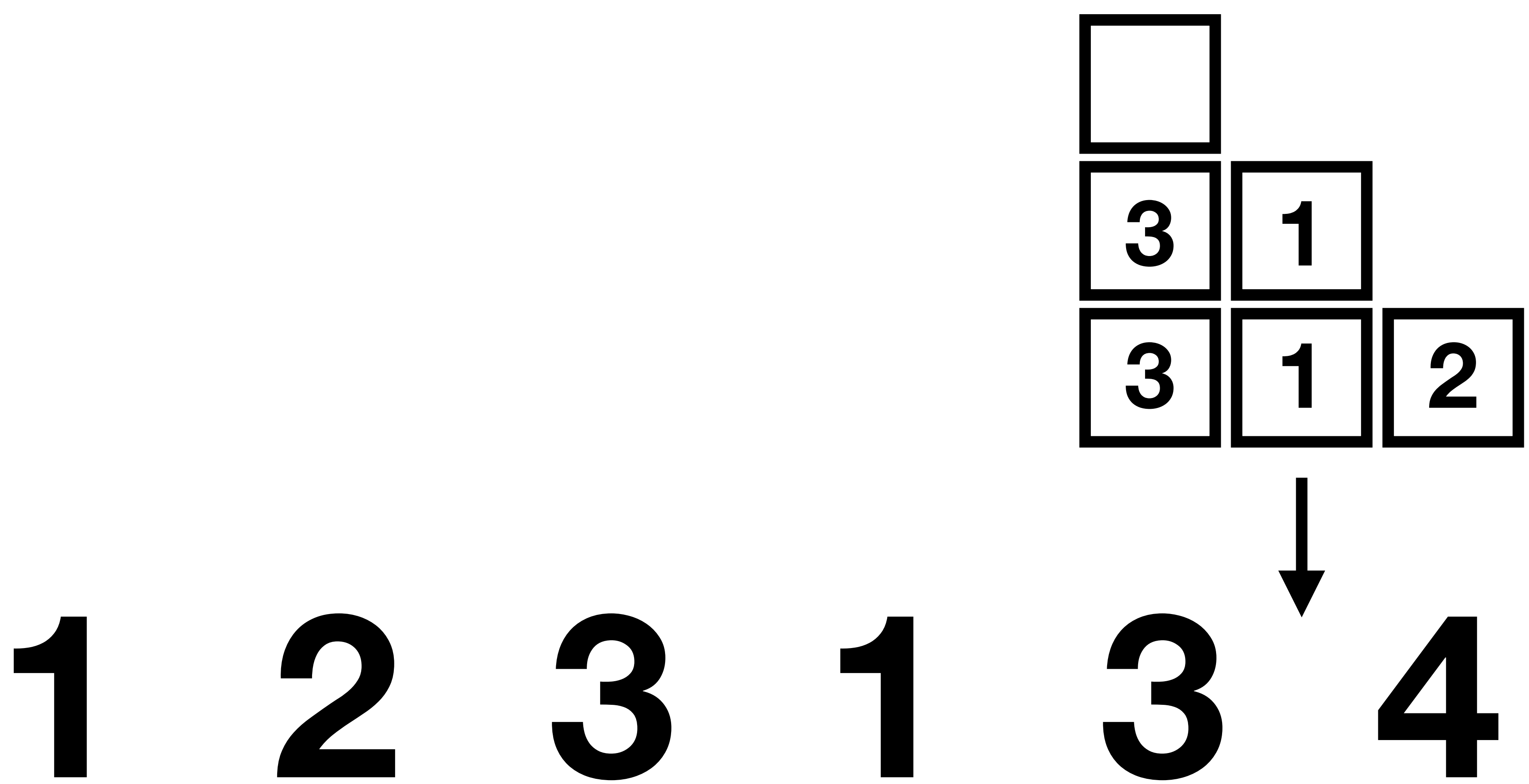
Single use register automata

There are at most 3 distinct letters



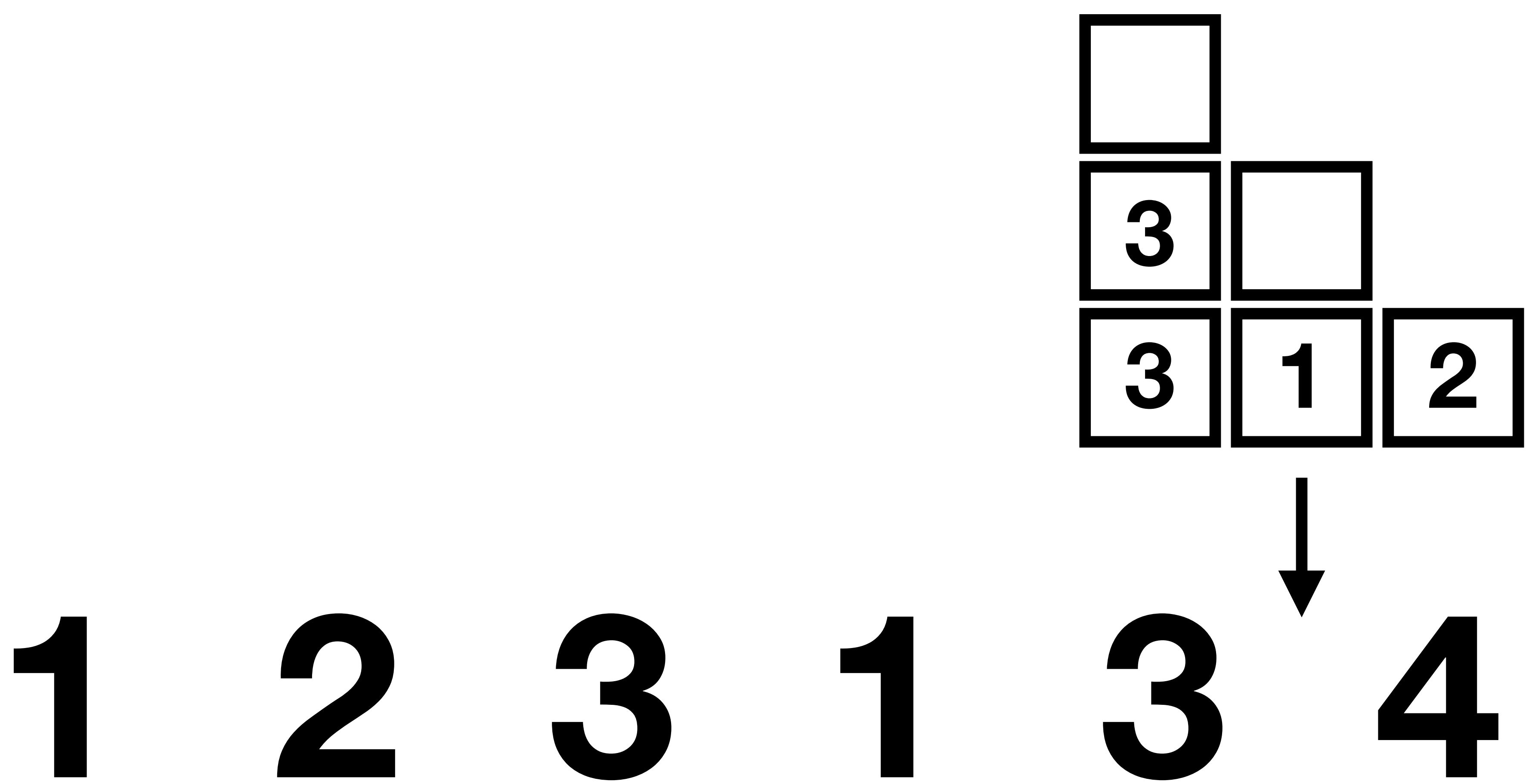
Single use register automata

There are at most 3 distinct letters



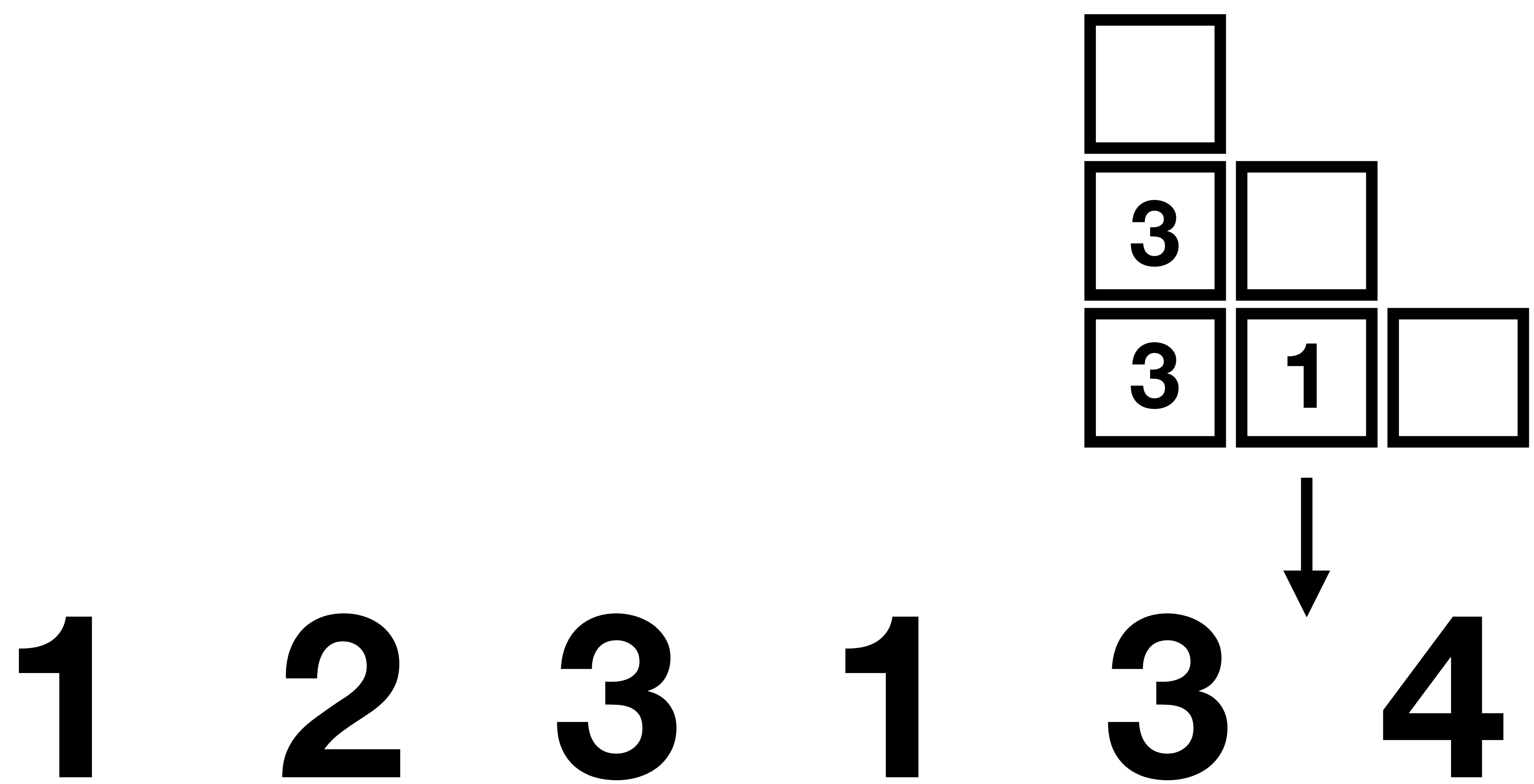
Single use register automata

There are at most 3 distinct letters



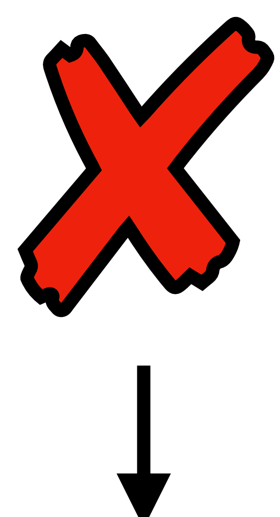
Single use register automata

There are at most 3 distinct letters



Single use register automata

There are at most 3 distinct letters

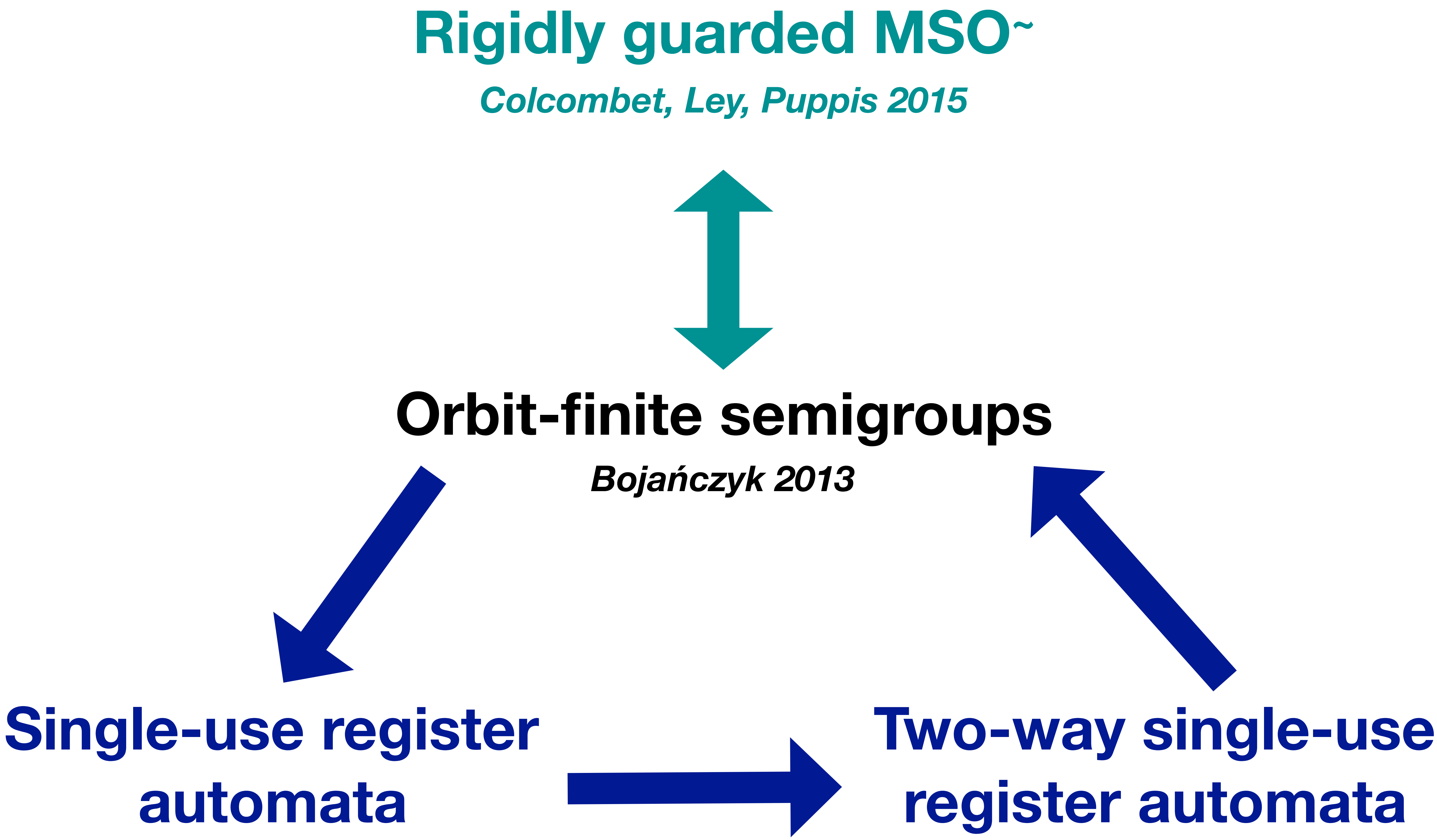


1 2 3 1 3 4

Single use register automata

There are at most 3 distinct letters

1 2 3 1 3 4


Rigidly guarded MSO~
Colcombet, Ley, Puppis 2015
Orbit-finite semigroups
Bojańczyk 2013
Single-use register automata
Two-way single-use register automata

Robustness — a case for the single use restriction

All the following models recognise different classes of languages

Multiple-use, deterministic register automata

Multiple-use, 2-way deterministic register automata

MSO~

Transducers

with atoms

An example

Remove repetitions from the input

1 1 2 3 3 3 2 2

An example

Remove repetitions from the input

1		2			3		2
1	1	2	3	3	3	2	2

An example

Remove repetitions from the input

1 2 3 2

1 1 2 3 3 3 2 2

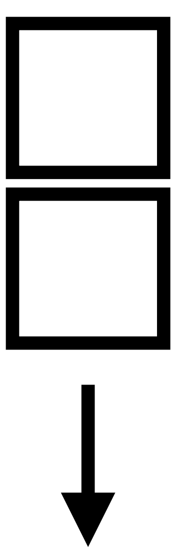
An example

Remove repetitions from the input

1 1 2 3 3 3 2 2

An example

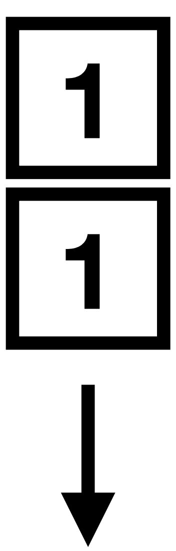
Remove repetitions from the input



1 1 2 3 3 3 2 2

An example

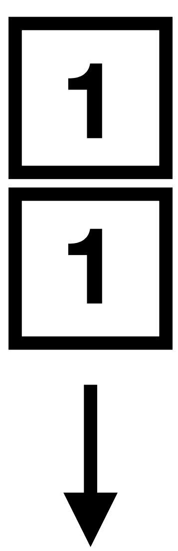
Remove repetitions from the input



1 1 2 3 3 3 2 2

An example

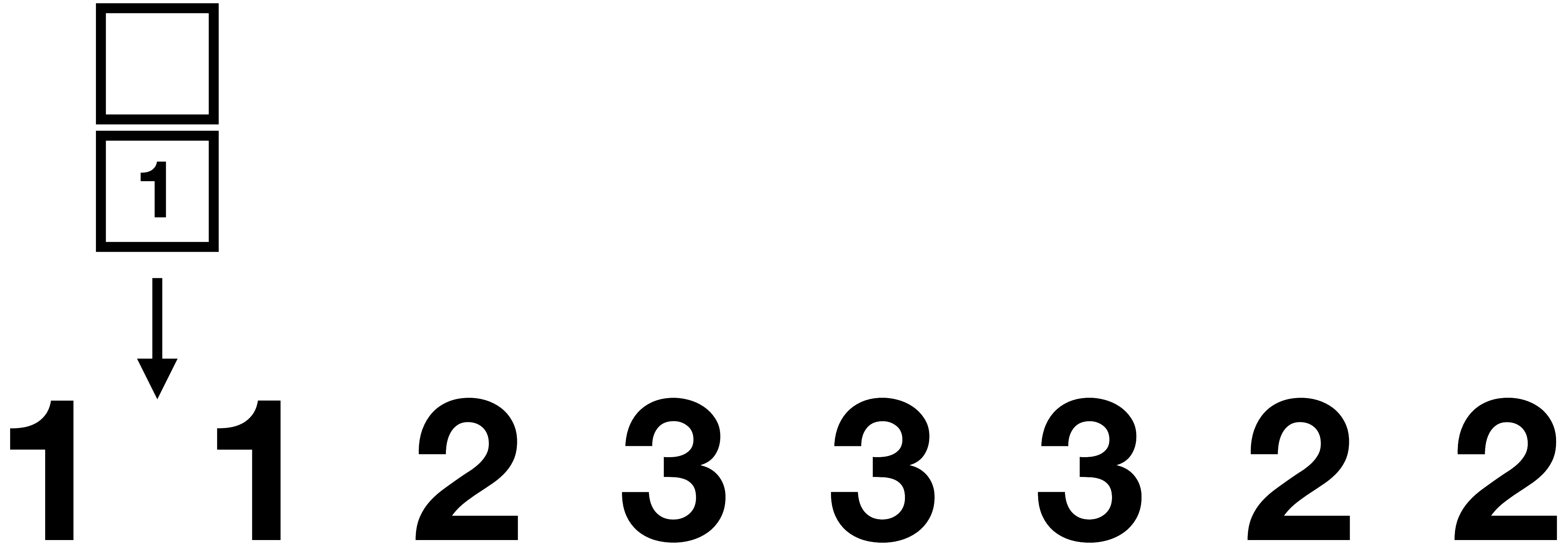
Remove repetitions from the input



1 1 2 3 3 3 2 2

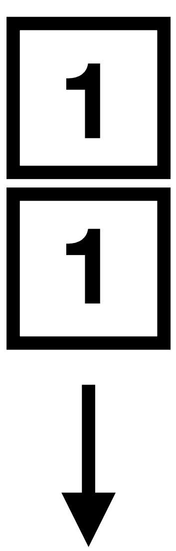
An example

Remove repetitions from the input



An example

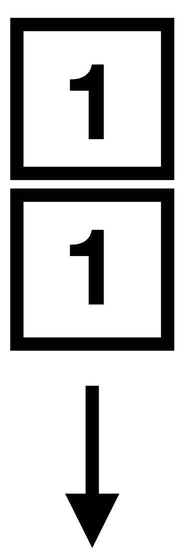
Remove repetitions from the input



1 1 2 3 3 3 2 2

An example

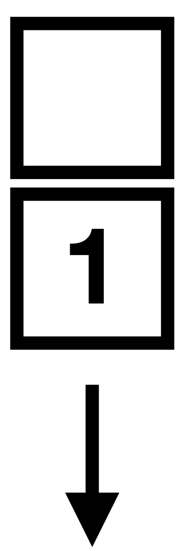
Remove repetitions from the input



1 1 2 3 3 3 2 2

An example

Remove repetitions from the input

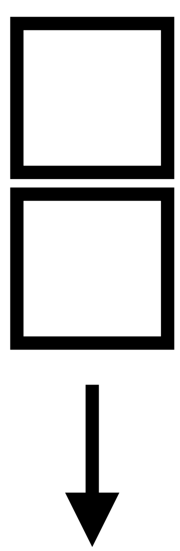


1 1 2 3 3 3 2 2

An example

Remove repetitions from the input

1

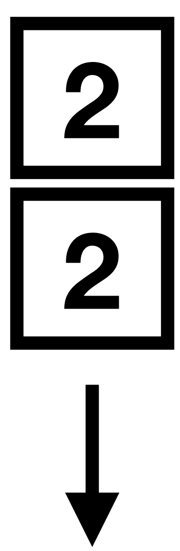


1 1 2 3 3 3 2 2

An example

Remove repetitions from the input

1

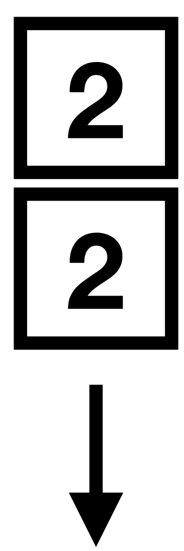


1 1 2 3 3 3 2 2

An example

Remove repetitions from the input

1

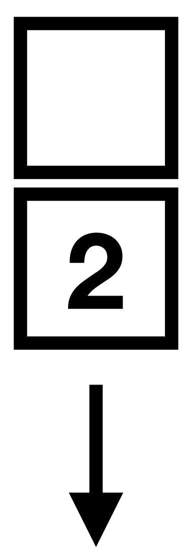


1 1 2 3 3 3 2 2

An example

Remove repetitions from the input

1

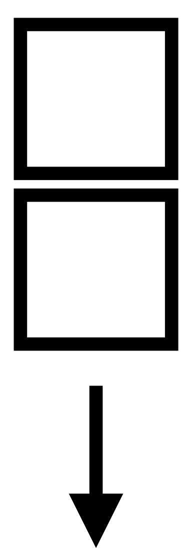


1 1 2 3 3 3 2 2

An example

Remove repetitions from the input

1 2

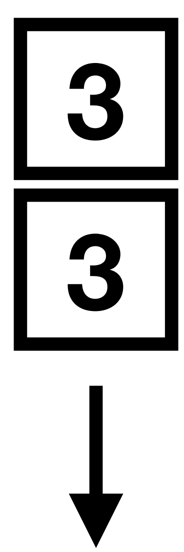


1 1 2 3 3 3 2 2

An example

Remove repetitions from the input

1 2

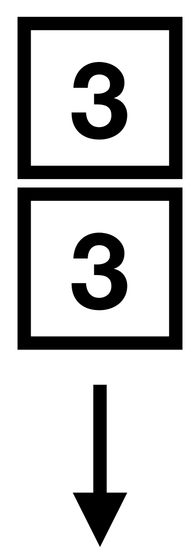


1 1 2 3 3 3 2 2

An example

Remove repetitions from the input

1 2

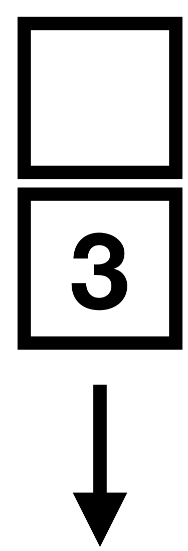


1 1 2 3 3 3 2 2

An example

Remove repetitions from the input

1 2

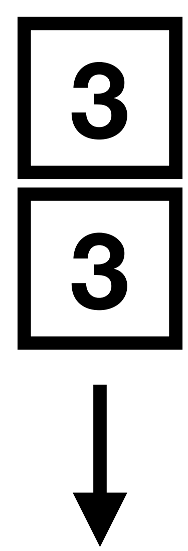


1 1 2 3 3 3 2 2

An example

Remove repetitions from the input

1 2



1 1 2 3 3 3 2 2

An example

Remove repetitions from the input

1 2

3

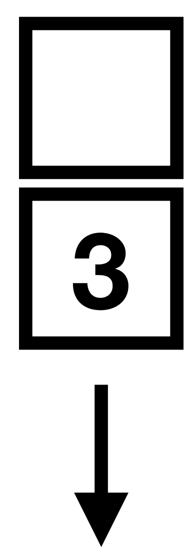
3

1 1 2 3 3 3 2 2

An example

Remove repetitions from the input

1 2

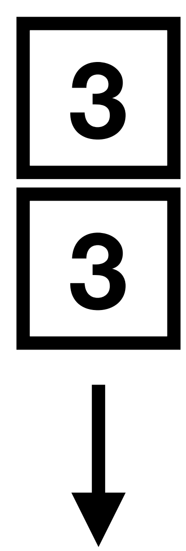


1 1 2 3 3 3 2 2

An example

Remove repetitions from the input

1 2

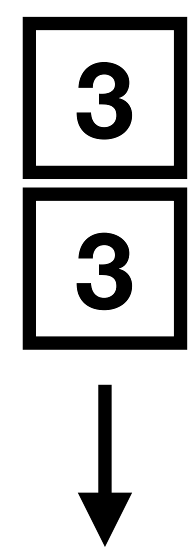


1 1 2 3 3 3 2 2

An example

Remove repetitions from the input

1 2

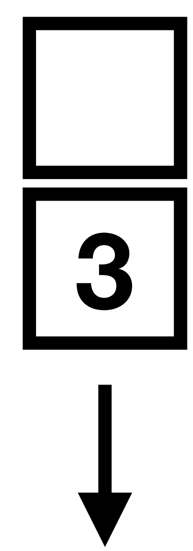


1 1 2 3 3 3 2 2

An example

Remove repetitions from the input

1 2

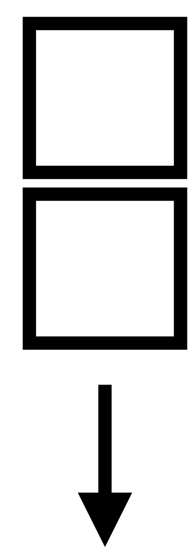


1 1 2 3 3 3 2 2

An example

Remove repetitions from the input

1 2 3

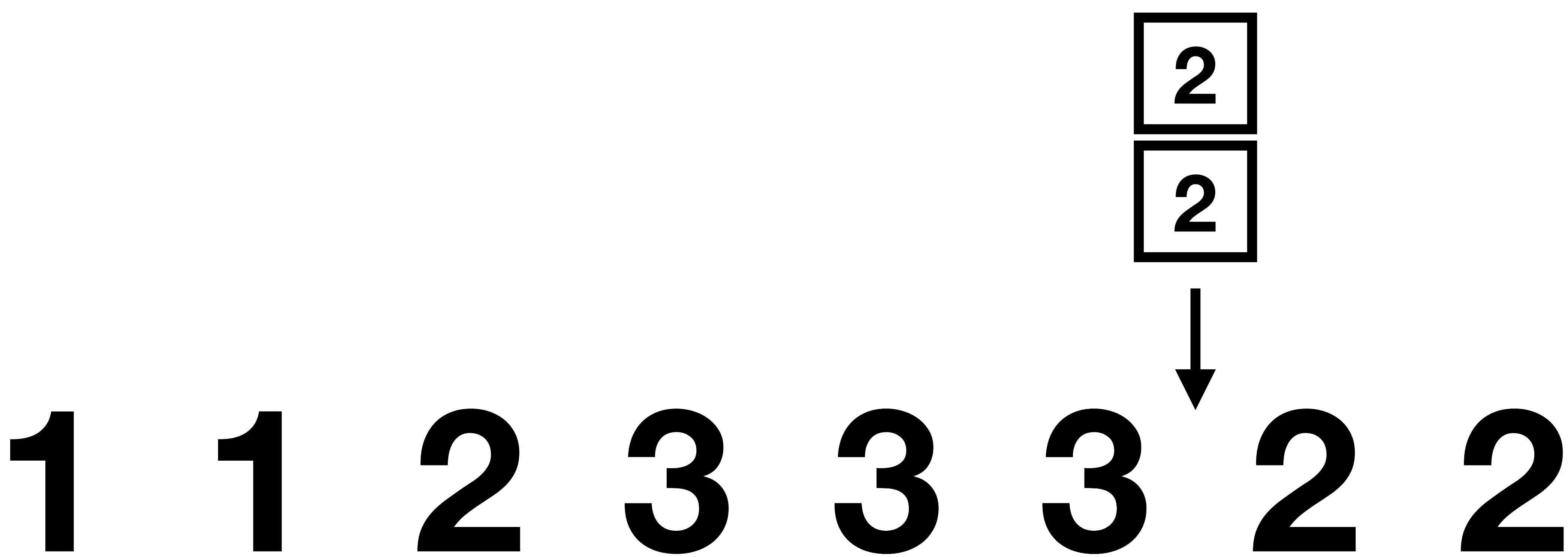


1 1 2 3 3 3 2 2

An example

Remove repetitions from the input

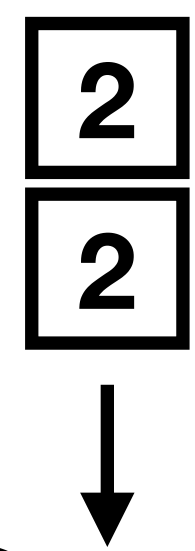
1 2 3



An example

Remove repetitions from the input

1 2 3

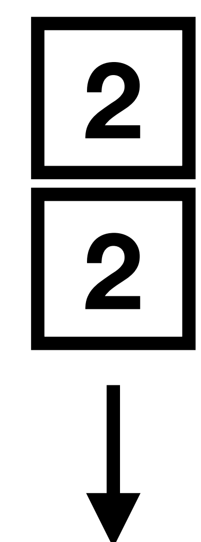


1 1 2 3 3 3 2 2

An example

Remove repetitions from the input

1 2 3

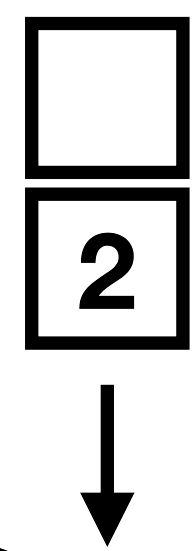


1 1 2 3 3 3 2 2

An example

Remove repetitions from the input

1 2 3

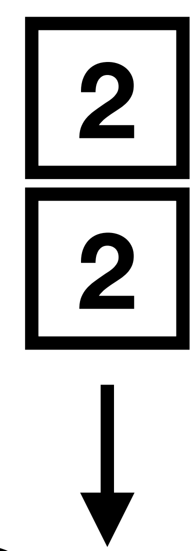


1 1 2 3 3 3 2 2

An example

Remove repetitions from the input

1 2 3



1 1 2 3 3 3 2 2

An example

Remove repetitions from the input

1 2 3

2

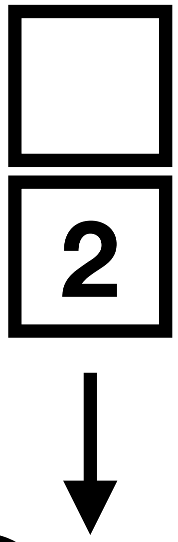
2

1 1 2 3 3 3 2 2

An example

Remove repetitions from the input

1 2 3 2



1 1 2 3 3 3 2 2

An example

Remove repetitions from the input

1 2 3 2

1 1 2 3 3 3 2 2

Krohn-Rhodes theorem

Prime functions

(Equivariant) homomorphism

Delay

Finite group on prefixes

Flip-flop on prefixes

Letter propagation

Prime functions

(Equivariant) homomorphism

Delay

Finite group on prefixes

Flip-flop on prefixes

Letter propagation

Prime functions

(Equivariant) homomorphism

An equivariant function: $f : \Sigma \to \Gamma^*$

Extended to: $f^* : \Sigma^* \to \Gamma^*$

Prime functions

(Equivariant) homomorphism

Double the letters:

1 2 3 2

Prime functions

(Equivariant) homomorphism

1 1 2 2 3 3 2 2

1 2 3 2

Prime functions

(Equivariant) homomorphism

1 1 2 2 3 3 2 2

Prime functions

(Equivariant) homomorphism

Delay

Finite group on prefixes

Flip-flop on prefixes

Letter propagation

Prime functions

Delay $\Sigma^* \to \left((\Sigma + \{\vdash, \dashv\})^2\right)^*$

Remove repetitions:

1 1 2 3 3 3 2 2

Prime functions

Delay $\Sigma^* \to \left((\Sigma + \{\vdash, \dashv\})^2\right)^*$

Remove repetitions:

$$\begin{matrix} \vdash & 1 & 1 & 2 & 3 & 3 & 3 & 2 & 2 \\ 1 & 1 & 2 & 3 & 3 & 3 & 2 & 2 & \dashv \end{matrix}$$

Prime functions

Delay $\Sigma^* \to \left((\Sigma + \{\vdash, \dashv\})^2\right)^*$

Remove repetitions:

1	2	3	2
2	3	2	$\dashv$

Prime functions

Delay $\Sigma^* \to \big((\Sigma + \{\vdash, \dashv\})^2\big)^*$

Remove repetitions:

1 2 3 2

Prime functions

Delay $\Sigma^* \to \big((\Sigma + \{\vdash, \dashv\})^2\big)^*$

Remove repetitions:

1 2 3 2

Prime functions

(Equivariant) homomorphism

Delay

Finite group on prefixes

Flip-flop on prefixes

Letter propagation

Prime functions

Finite group on prefixes $(\Sigma \times G)^* \to (\Sigma \times G)^*$

Prime functions

Finite group on prefixes $(\Sigma \times G)^* \to (\Sigma \times G)^*$

Remove letters from odd positions:

1 9 3 9 2 7

Prime functions

Finite group on prefixes $(\Sigma \times G)^* \to (\Sigma \times G)^*$

Remove letters from odd positions:

1 1 1 1 1 1 $(\mathbb{Z}_2)$

1 9 3 9 2 7

Prime functions

Finite group on prefixes $(\Sigma \times G)^* \to (\Sigma \times G)^*$

Remove letters from odd positions:

1 0 1 0 1 0

1 9 3 9 2 7

Prime functions

Finite group on prefixes $(\Sigma \times G)^* \to (\Sigma \times G)^*$

Remove letters from odd positions:

0	0	0
9	**9**	**7**

Prime functions

Finite group on prefixes $(\Sigma \times G)^* \to (\Sigma \times G)^*$

Remove letters from odd positions:

9 9 7

Prime functions

(Equivariant) homomorphism

Delay

Finite group on prefixes

Flip-flop on prefixes

Letter propagation

Prime functions

Flip-flop monoid on prefixes $(\Sigma \times F)^* \to (\Sigma \times F)^*$

Prime functions

Flip-flop monoid on prefixes $(\Sigma \times F)^* \to (\Sigma \times F)^*$

$$F = \{1, a, b\}$$

$$1a = aa = ba = a$$

$$1b = bb = ab = b$$

Prime functions

Flip-flop monoid on prefixes $(\Sigma \times F)^* \to (\Sigma \times F)^*$

Remove everything after the first repetition:

1 2 5 7 7 7 8 8 9

Prime functions

Flip-flop monoid on prefixes $(\Sigma \times F)^* \to (\Sigma \times F)^*$

Remove everything after the first repetition:

$\vdash$ 1 2 5 7 7 7 8 8 9

1 2 5 7 7 7 8 8 9 $\dashv$

Prime functions

Flip-flop monoid on prefixes $(\Sigma \times F)^* \to (\Sigma \times F)^*$

Remove everything after the first repetition:

1 1 1 1 a a 1 a 1 1

⊢ 1 2 5 7 7 7 8 8 9

1 2 5 7 7 7 8 8 9 ⊣

Prime functions

Flip-flop monoid on prefixes $(\Sigma \times F)^* \to (\Sigma \times F)^*$

Remove everything after the first repetition:

$1\ 1\ 1\ 1\ a\ a\ a\ a\ a\ a$

$\vdash$ **1 2 5 7 7 7 8 8 9**

1 2 5 7 7 7 8 8 9 $\dashv$

Prime functions

Flip-flop monoid on prefixes $(\Sigma \times F)^* \to (\Sigma \times F)^*$

Remove everything after the first repetition:

1 1 1

1 2 5

2 5 7

Prime functions

Flip-flop monoid on prefixes $(\Sigma \times F)^* \to (\Sigma \times F)^*$

Remove everything after the first repetition:

Original Krohn-Rhodes theorem

Every function *f* recognised by a one-way transducer iff it is an element of *(prime)**

Original Krohn-Rhodes theorem

Every function *f* recognised by a one-way transducer can also be expressed as an element of *(prime)**

$$\texttt{remove\_repetitions} = f^* \circ \texttt{delay}$$

$$f(\langle x, y \rangle) = \begin{cases} y & \text{if } x \neq y \\ \epsilon & \text{otherwise} \end{cases}$$

Prime functions

(Equivariant) homomorphism

Delay

Finite group on prefixes

Flip-flop on prefixes

Letter propagation

Prime functions

Letter propagataion $(\Sigma \times P\{\uparrow, \downarrow\})^* \to (\Sigma \times (\{.\} + \Sigma))^*$

Prime functions

Letter propagation $(\Sigma \times P\{\uparrow,\downarrow\})^* \to (\Sigma \times (\{.\} + \Sigma))^*$

Operations on 1 register:

$\uparrow$ Read value

$\downarrow$ Output value

Prime functions

Letter propagation $(\Sigma \times P\{\uparrow, \downarrow\})^* \to (\Sigma \times (\{.\} + \Sigma))^*$

Operations on 1 register:

$\uparrow$ Read value

$\downarrow$ Output value

Subject to single-use restriction

Prime functions

Letter propagation $(\Sigma \times P\{\uparrow, \downarrow\})^* \to (\Sigma \times (\{.\} + \Sigma))^*$

Change the last letter to the first letter:

1 2 5 7 7 9

Prime functions

Letter propagation $(\Sigma \times P\{\uparrow,\downarrow\})^* \to (\Sigma \times (\{.\} + \Sigma))^*$

Change the last letter to the first letter:

$\vdash$ **1 2 5 7 7 9**

1 2 5 7 7 9 $\dashv$

Prime functions

Letter propagation $(\Sigma \times P\{\uparrow,\downarrow\})^* \to (\Sigma \times (\{.\} + \Sigma))^*$

Change the last letter to the first letter:

$\downarrow$ $\qquad\qquad\qquad\qquad\qquad\qquad$ $\uparrow$

$\vdash$ **1 2 5 7 7 9**

1 2 5 7 7 9 $\dashv$

Prime functions

Letter propagation $(\Sigma \times P\{\uparrow, \downarrow\})^* \to (\Sigma \times (\{.\} + \Sigma))^*$

Change the last letter to the first letter:

$\vdash$

1

$\vdash$ **1 2 5 7 7 9**

1 2 5 7 7 9 $\dashv$

Prime functions

Letter propagation $(\Sigma \times P\{\uparrow, \downarrow\})^* \to (\Sigma \times (\{.\} + \Sigma))^*$

Change the last letter to the first letter:

1

1 2 5 7 7

Prime functions

Letter propagation $(\Sigma \times P\{\uparrow, \downarrow\})^* \to (\Sigma \times (\{.\} + \Sigma))^*$

Change the last letter to the first letter:

1 2 5 7 7 1

Krohn-Rhodes theorem with atoms

Every function *f* recognised by a single use, one-way iff it is an element of $(\textit{prime} + \textit{letter propagation})^*$

Two more prime functions

(Equivariant) homomorphism
Delay
Finite group on prefixes
Flip-flop on prefixes

Letter propagation

Iterated reverse
Iterated duplicate

Prime functions

Iterated reverse $(\Sigma + \{\#\})^* \to (\Sigma + \{\#\})^*$

1 2 5 # 7 9 # 9

Prime functions

Iterated reverse $(\Sigma + \{\#\})^* \to (\Sigma + \{\#\})^*$

5 2 1 # 9 7 # 9

1 2 5 # 7 9 # 9

Prime functions

Iterated reverse $(\Sigma + \{\#\})^* \to (\Sigma + \{\#\})^*$

521#97#9

Prime functions

Iterated duplicate $(\Sigma + \{\#\})^* \to (\Sigma + \{\#\})^*$

125#79#9

Prime functions

Iterated duplicate $(\Sigma + \{\#\})^* \to (\Sigma + \{\#\})^*$

1 2 5 1 2 5 # 7 9 7 9 # 9
1 2 5 # 7 9 # 9

Prime functions

Iterated duplicate $(\Sigma + \{\#\})^* \to (\Sigma + \{\#\})^*$

1 2 5 1 2 5 # 7 9 7 9 # 9

Two more prime functions

(Equivariant) homomorphism
Delay
Finite group on prefixes
Flip-flop on prefixes

Letter propagation

Iterated reverse
Iterated duplicate

Two-way Krohn-Rhodes theorem with atoms

single use, two way register transducer

=

*(two-way prime + letter propagation)**

Two corollaries

One-way single-use register automata are closed under compositions

Two-way single-use register automata are closed under compositions

One more corollary

All of the following recognise the same class of transductions:

(Two-way prime + letter propagation)*
original model by Krohn, Rhodes 1963

Two-way, single-use register transducers
original model by Shepherdson 1959

String streaming, single-use register transducers
original model by Alur, Cerny 2010

Regular list functions
original model by Bojańczyk, Daviaud, Krishna 2018

The general picture

1. (Two-way prime + letter propagation)*
2. Two-way, single-use register transducers
3. String streaming, single-use register transducers
4. Regular list functions

1. (Prime + reversed flip-flop + letter propagation + reversed letter propagation)*

1. (Prime + letter propagation)*
2. One-way, single-use register transducers

The general picture

1. (Two-way prime + letter propagation)*
2. Two-way, single-use register transducers
3. String streaming, single-use register transducers
4. Regular list functions
5. Rigid MSO~ transductions

1. (Prime + reversed flip-flop + letter propagation + reversed letter propagation)*

1. (Prime + letter propagation)*
2. One-way, single-use register transducers

The end