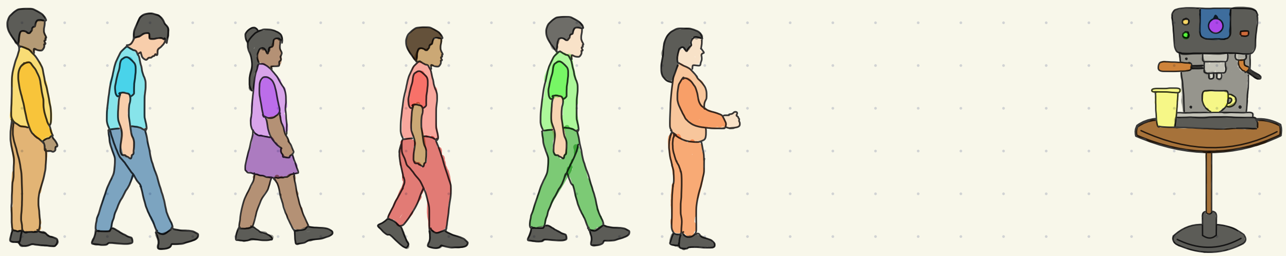
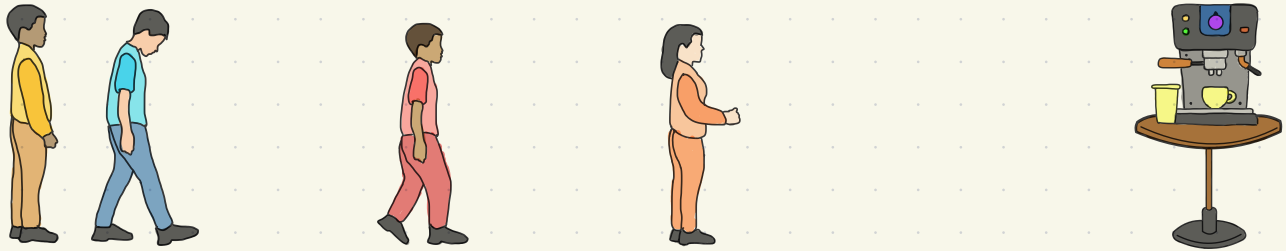
Priority Downward Closures

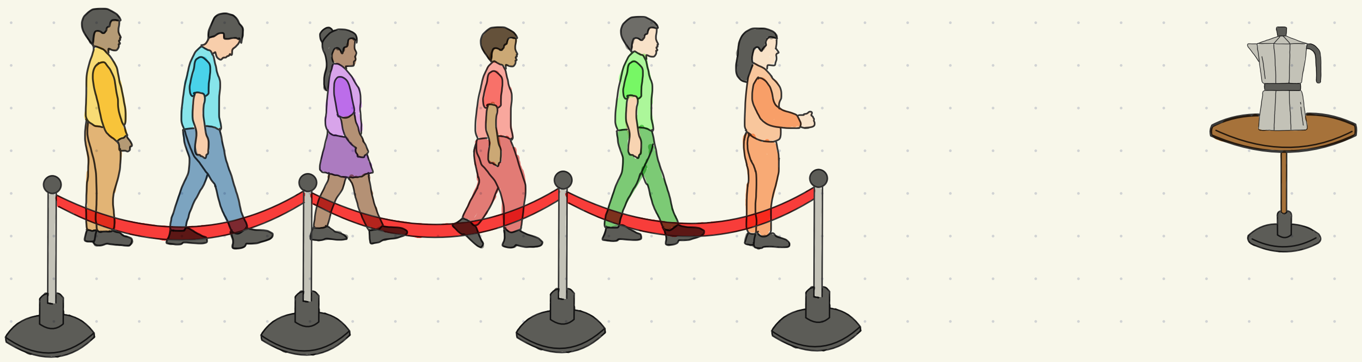
CONCUR'23

Ashwani Anand
Georg Zetzsche

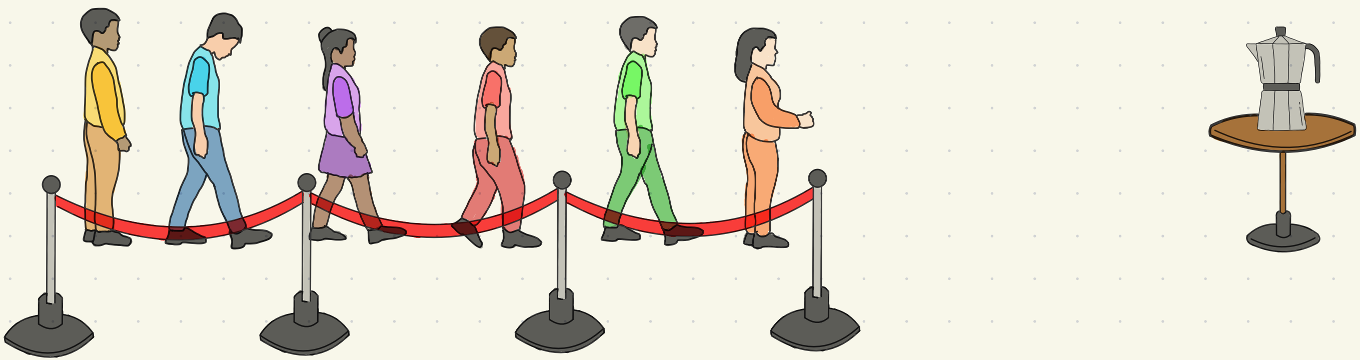




Will the coffee machine go to a bad state?

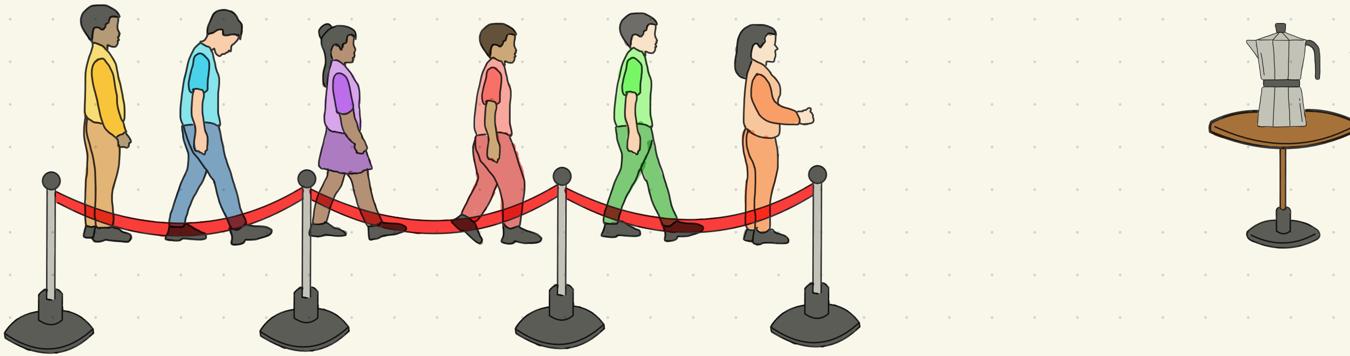


[Haines] The system can be overapproximated by a simpler system.



[Haines] The system can be overapproximated by a simpler system.

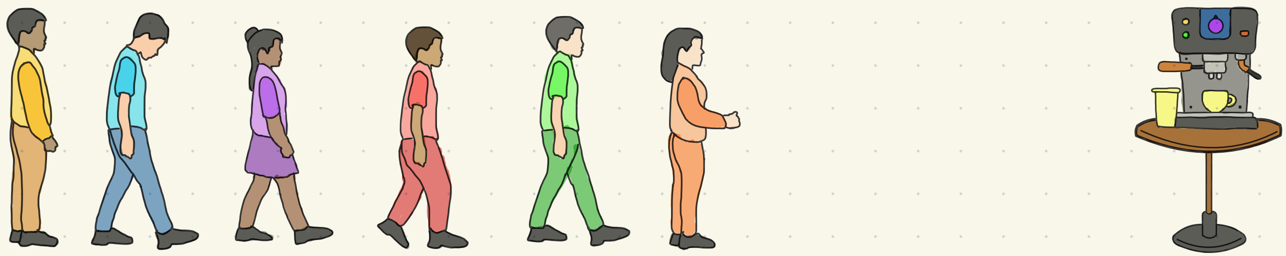
If a bad state is not reachable in new system it can not be reached in original system.

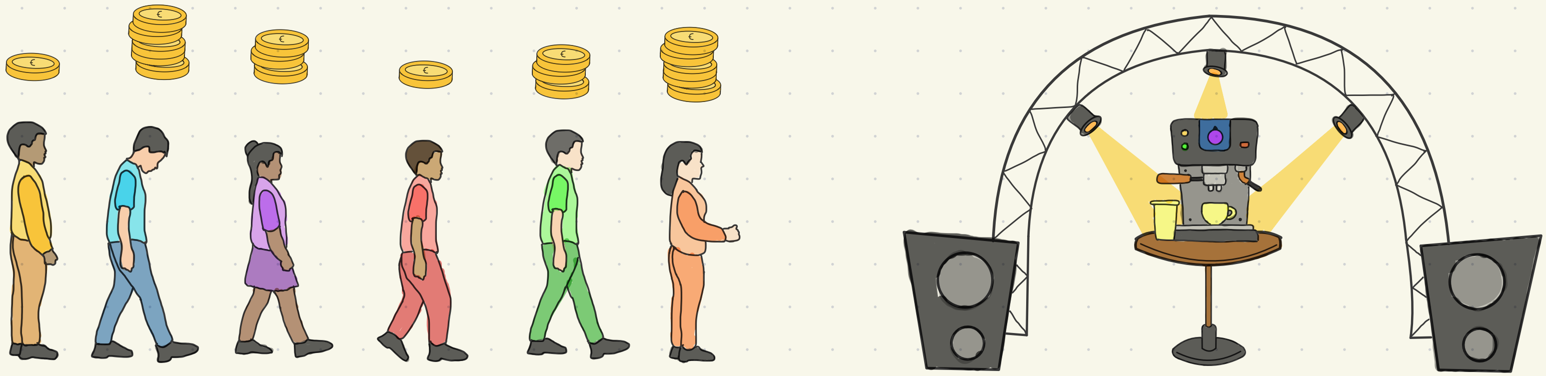


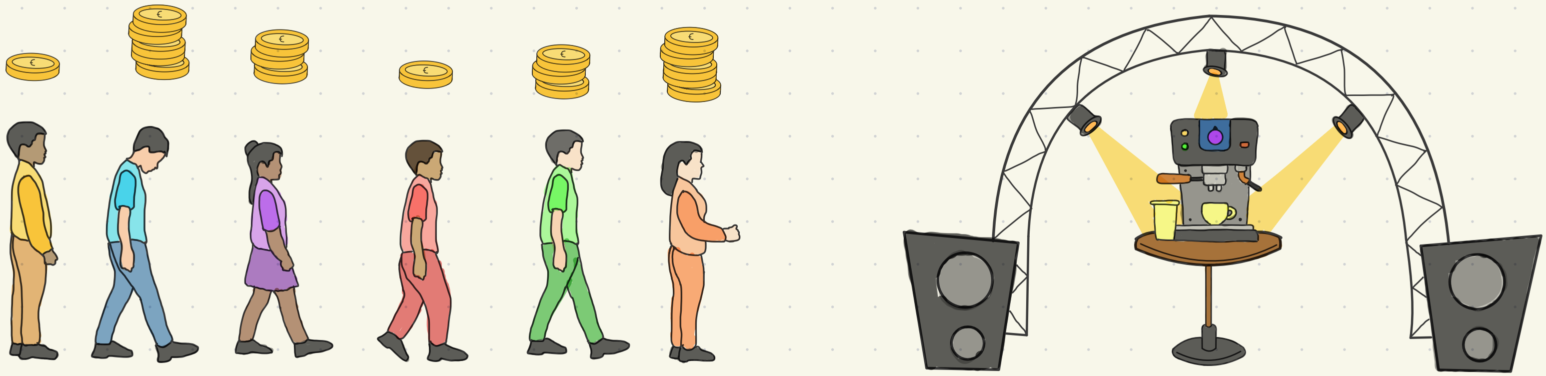
Might not be able to find it.

[Haines] The system can be overapproximated by a simpler system.

If a bad state is not reachable in new system it can not be reached in original system.



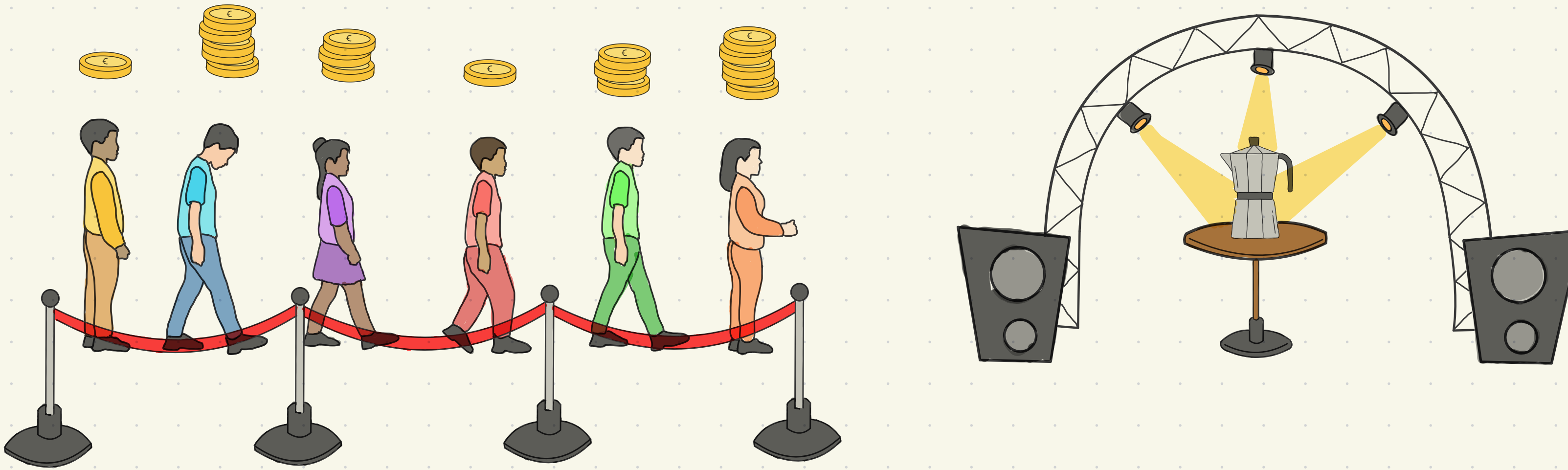




Can this system be overapproximated?

Can this system be overapproximated?

[This talk] Yes! For pushdown machines.

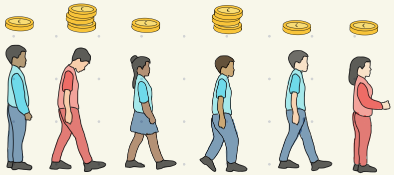


Can this system be overapproximated?

[This talk] Yes! For pushdown machines.

Precap

Block Order

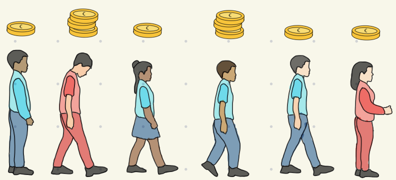


Simple Machines

Pushdown Machines

Precap

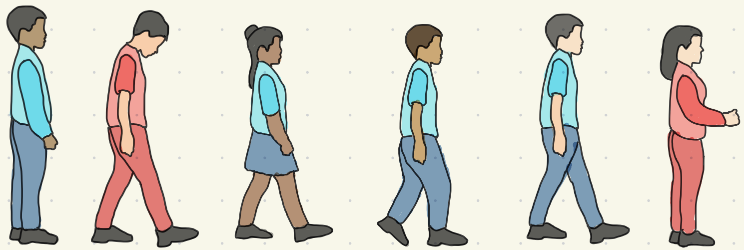
Block Order



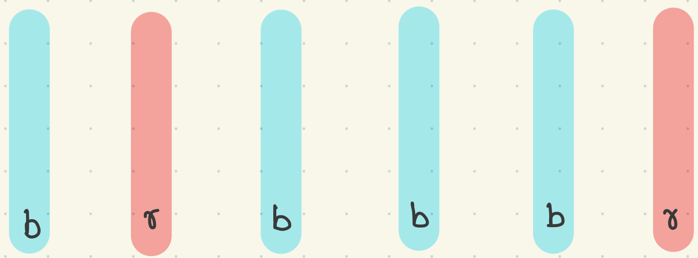
Simple Machines

Pushdown Machines

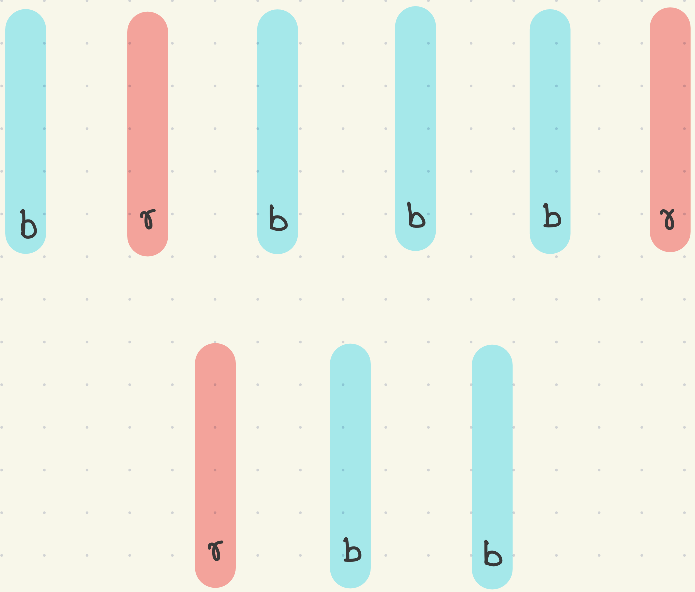
Subword Order



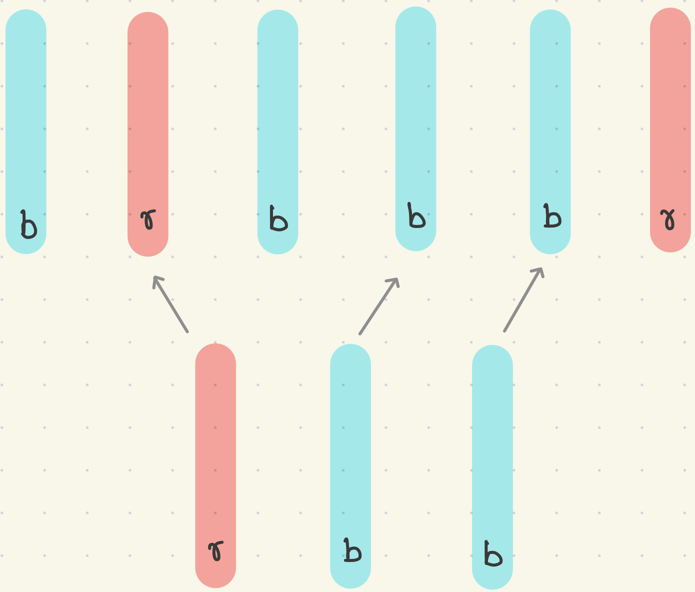
Subword Order



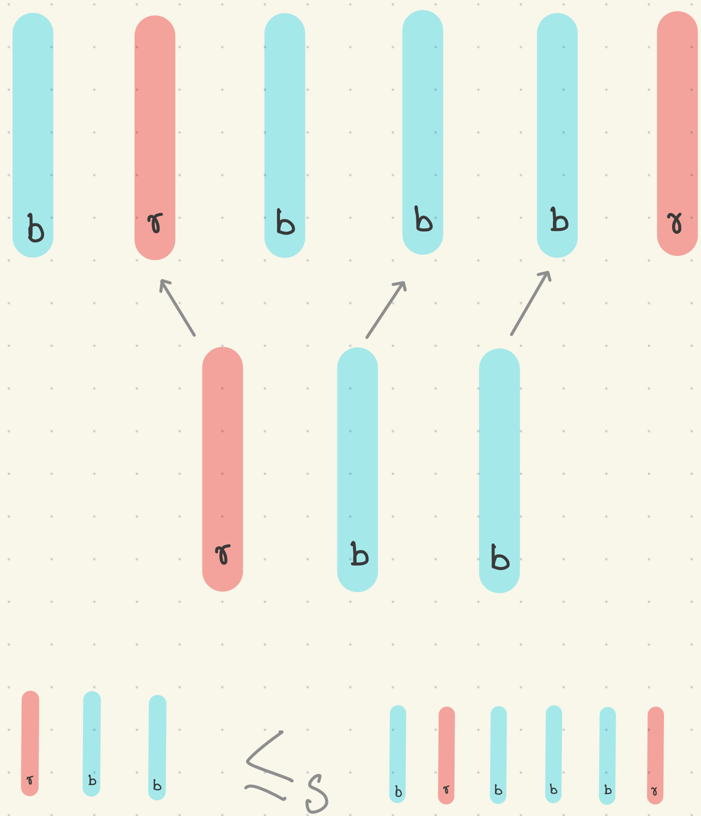
Subword Order



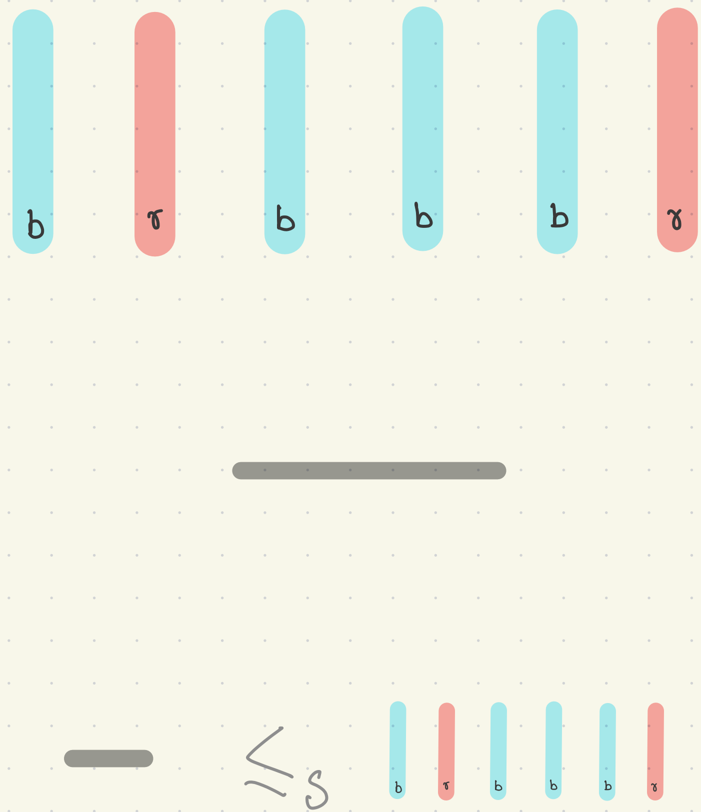
Subword Order



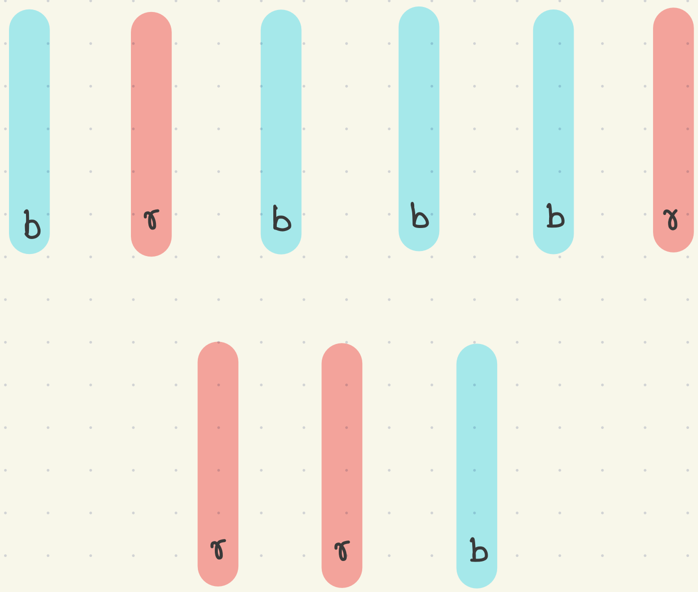
Subword Order



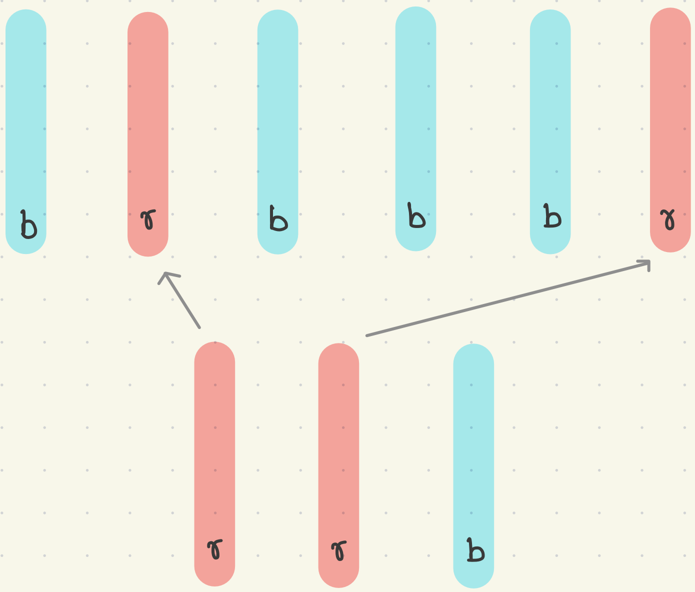
Subword Order



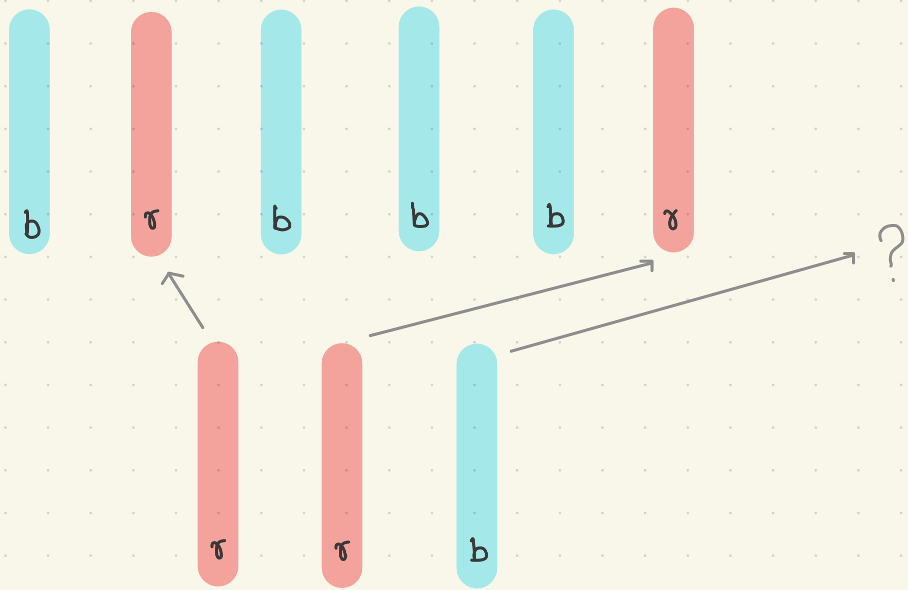
Subword Order



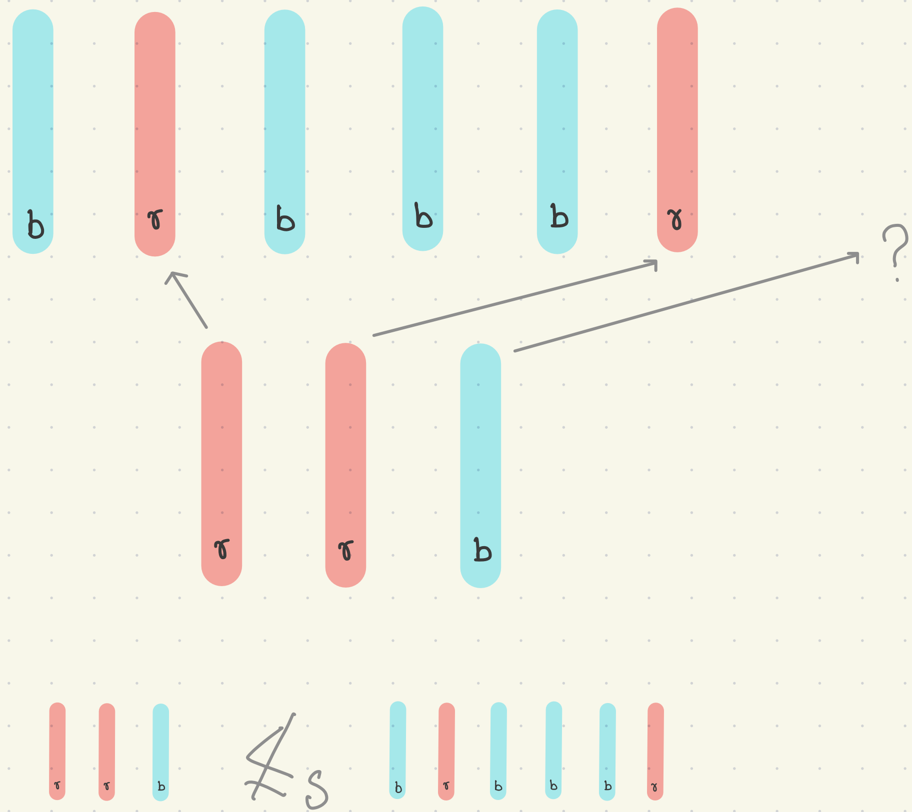
Subword Order



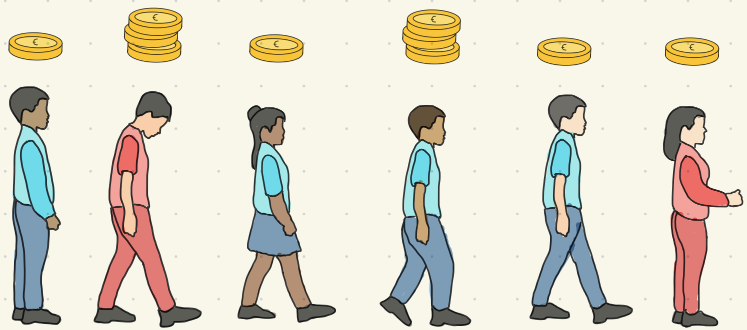
Subword Order



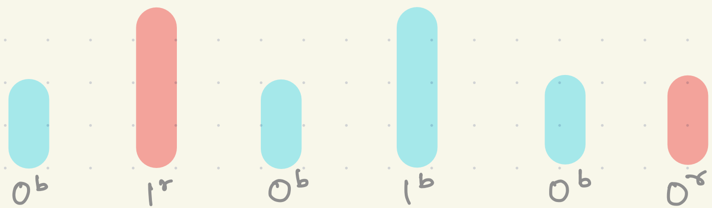
Subword Order



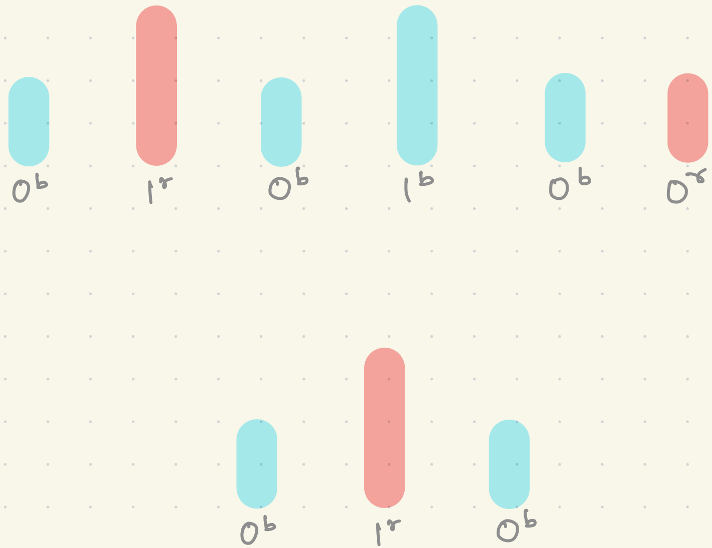
Block Order



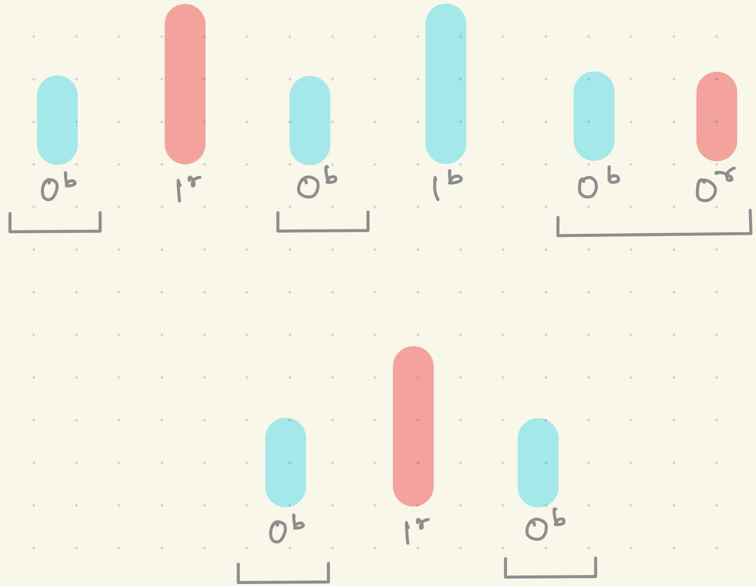
Block Order



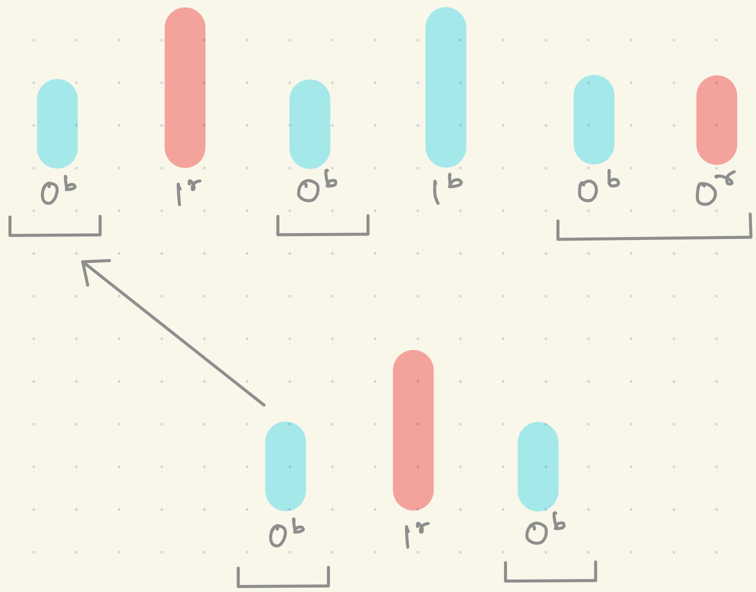
Block Order



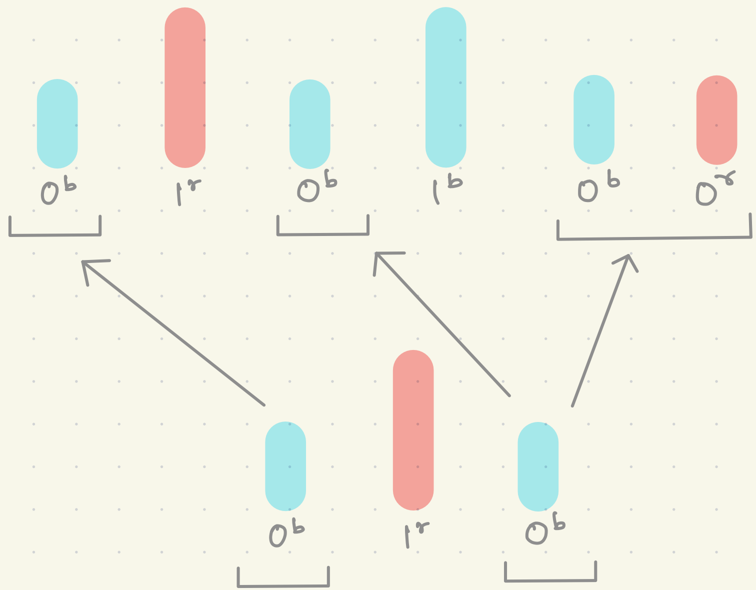
Block Order



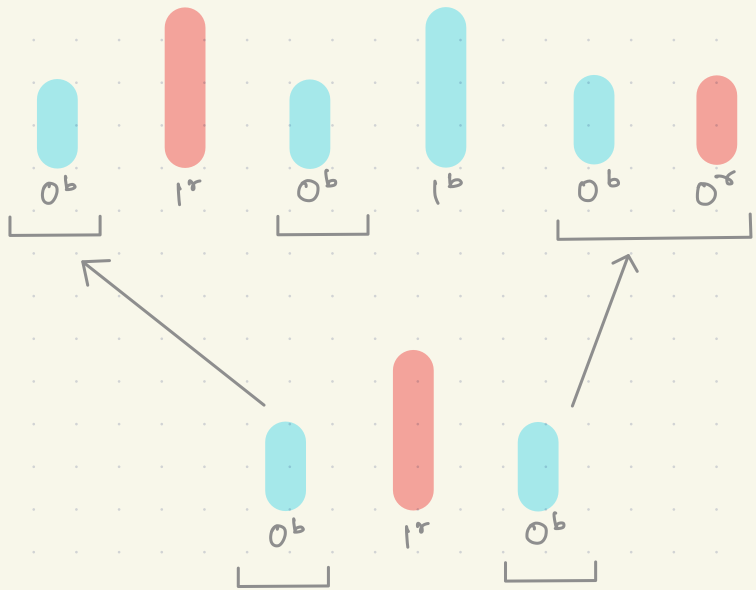
Block Order



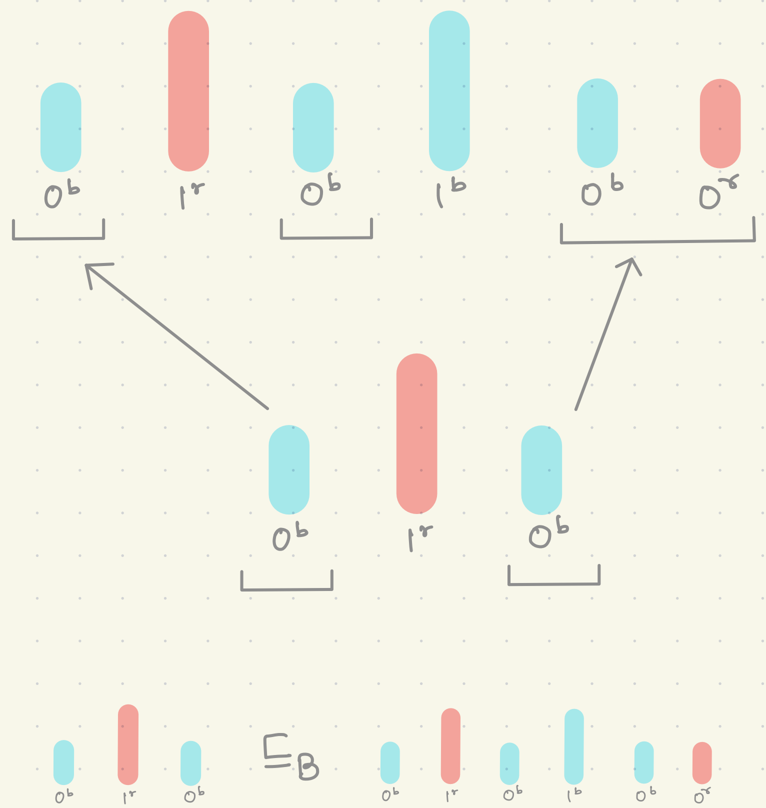
Block Order



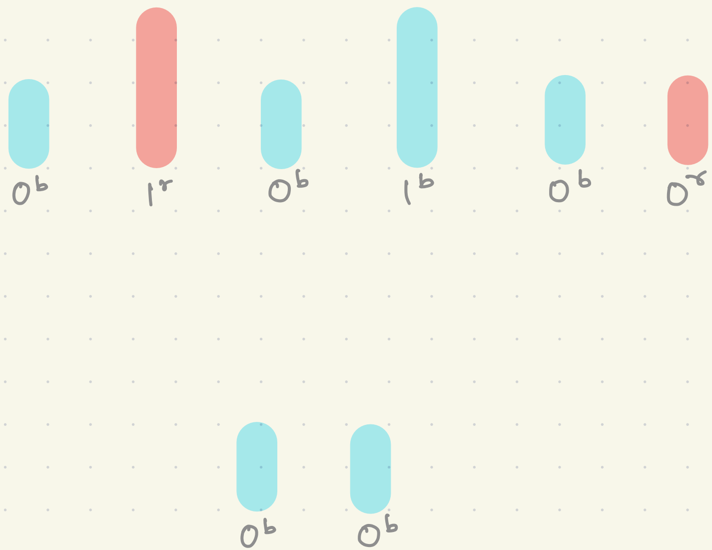
Block Order



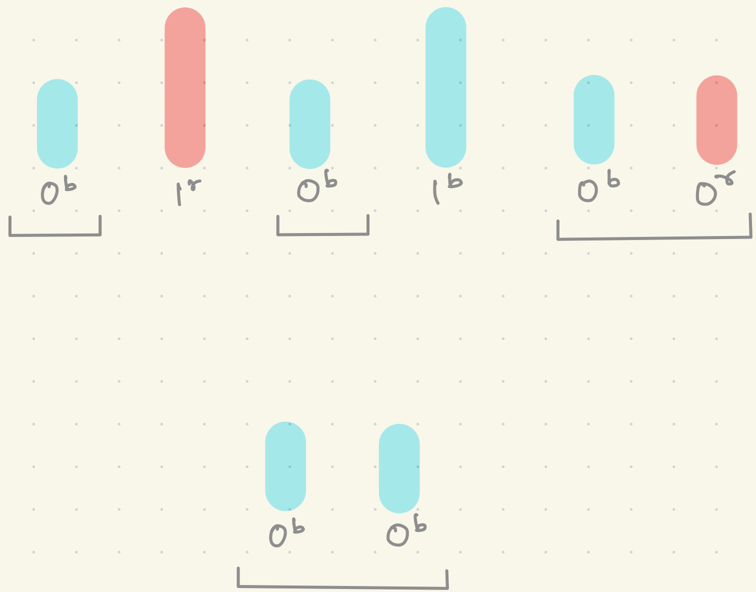
Block Order



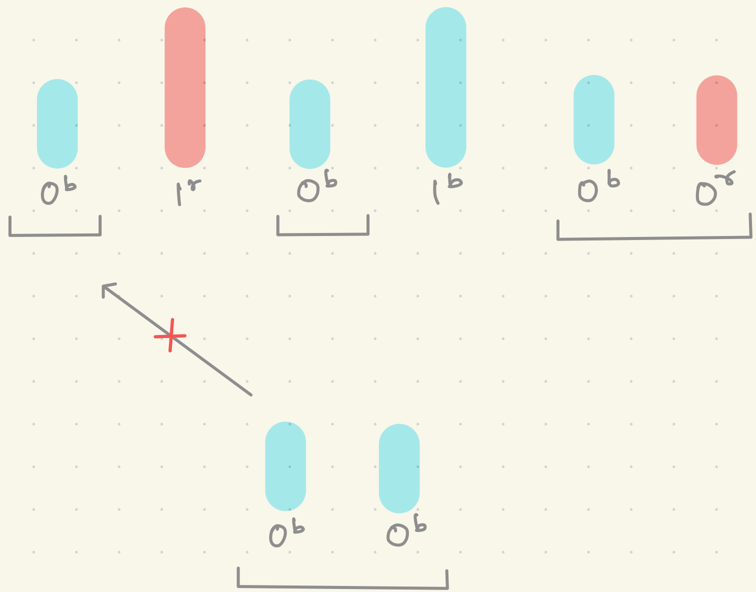
Block Order



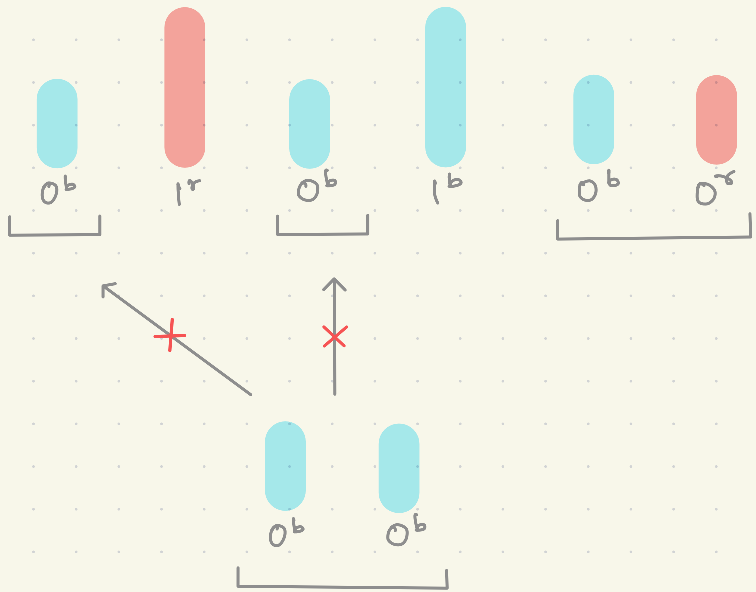
Block Order



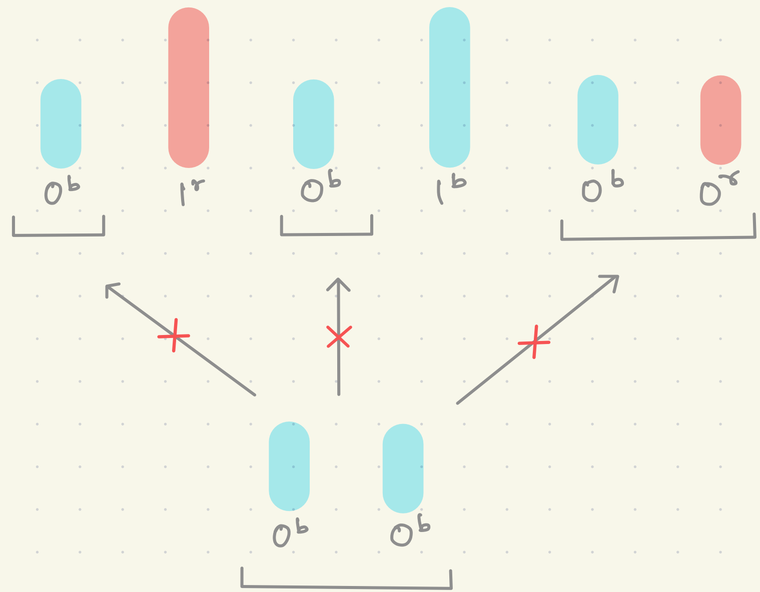
Block Order



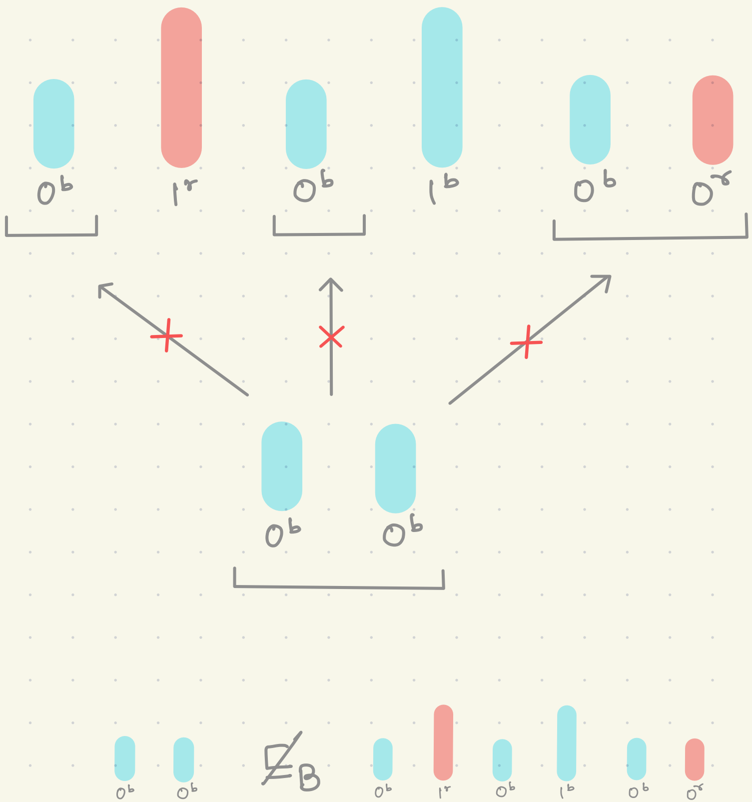
Block Order



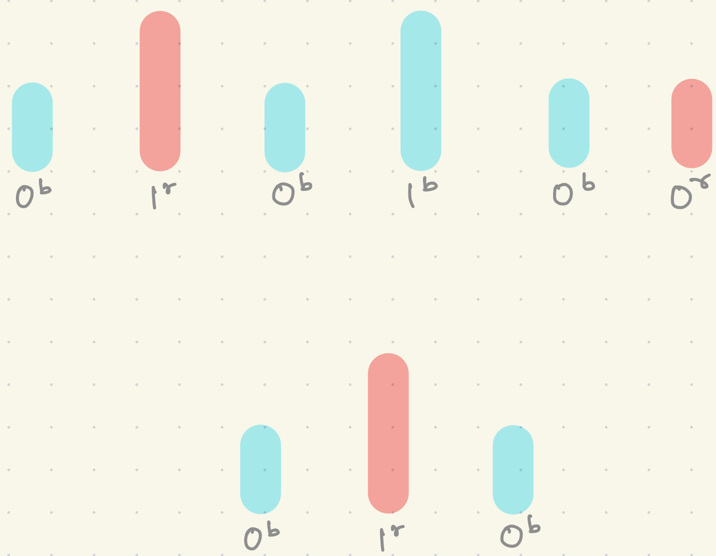
Block Order



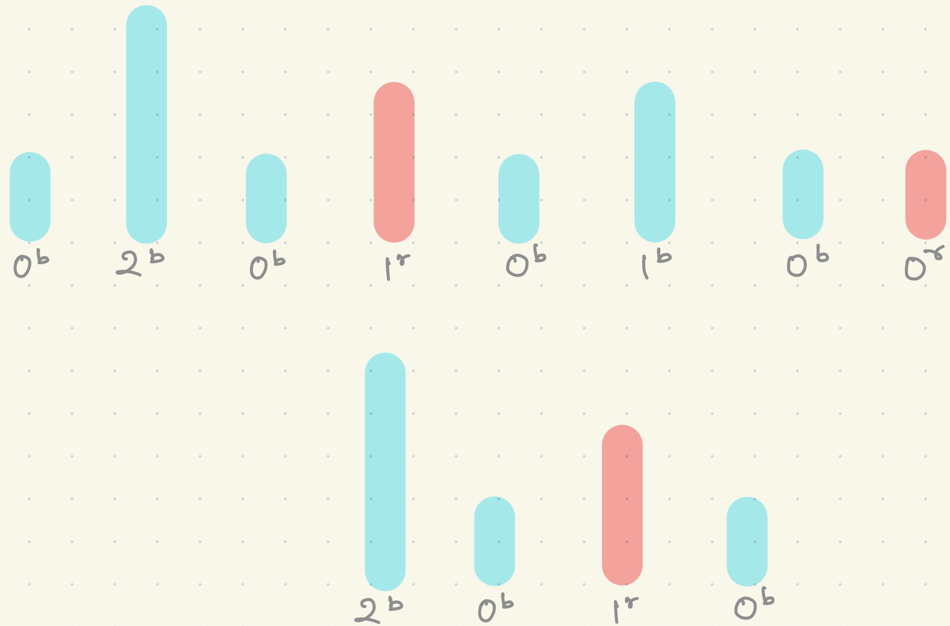
Block Order



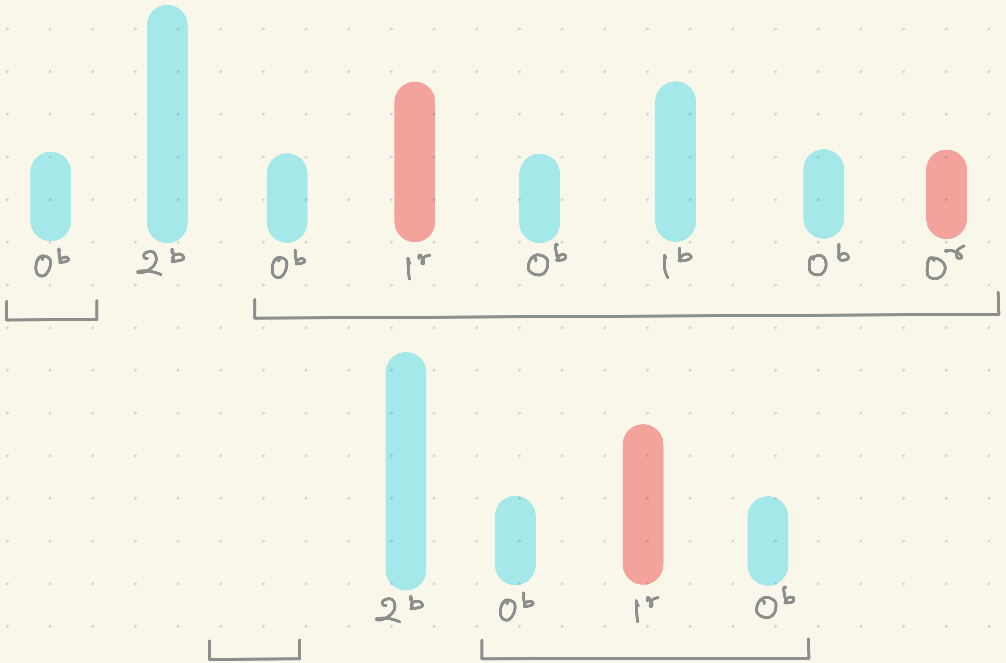
Block Order



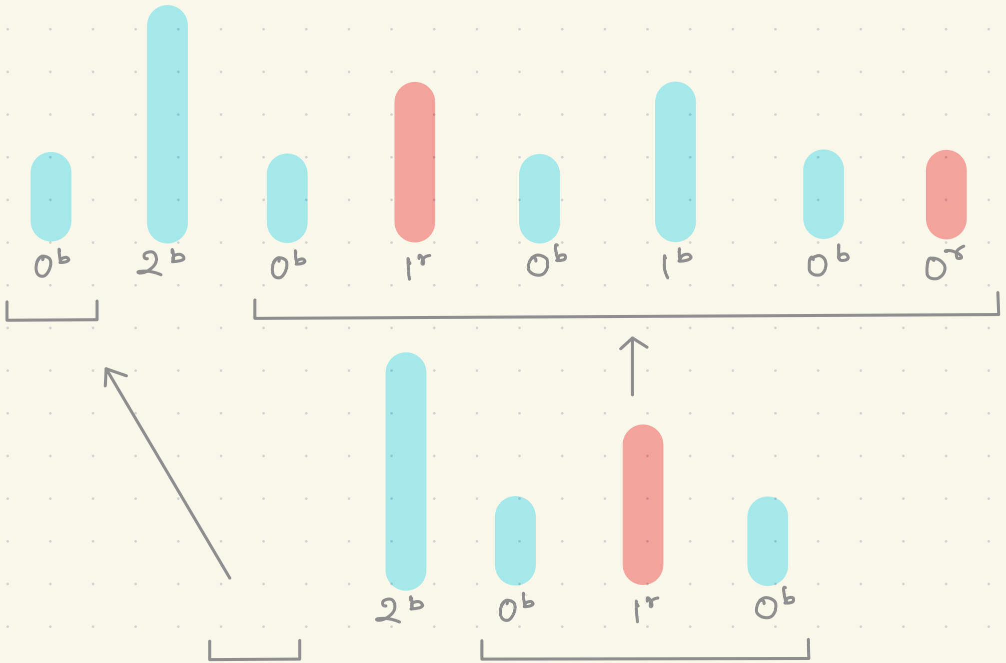
Block Order



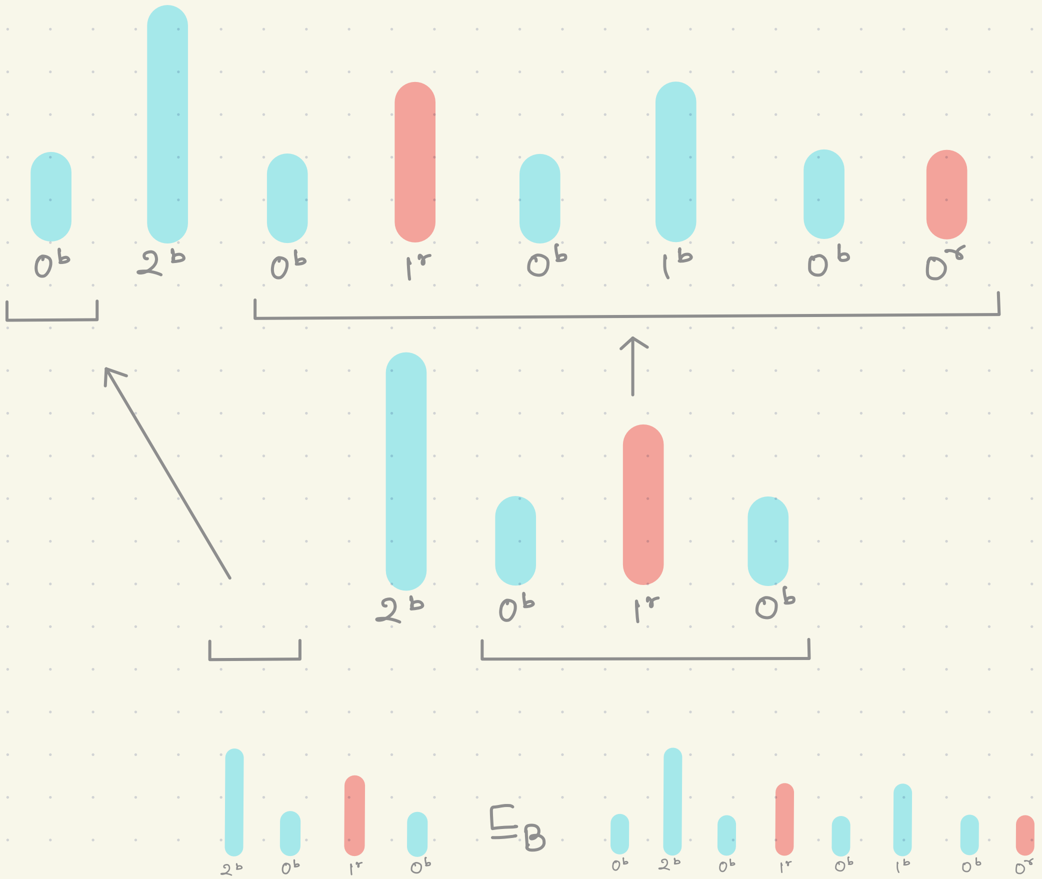
Block Order



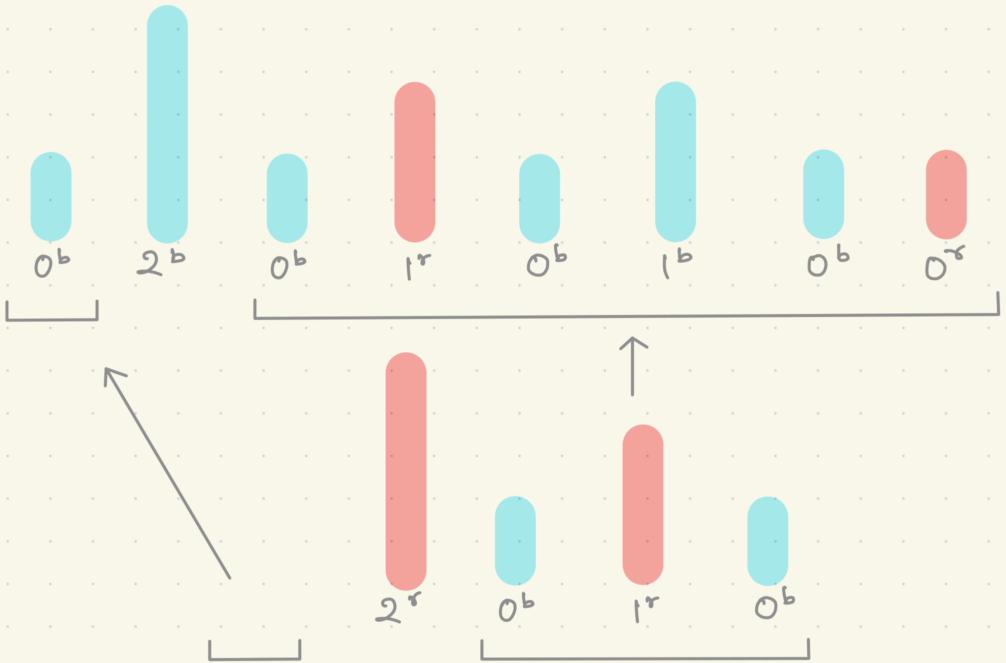
Block Order



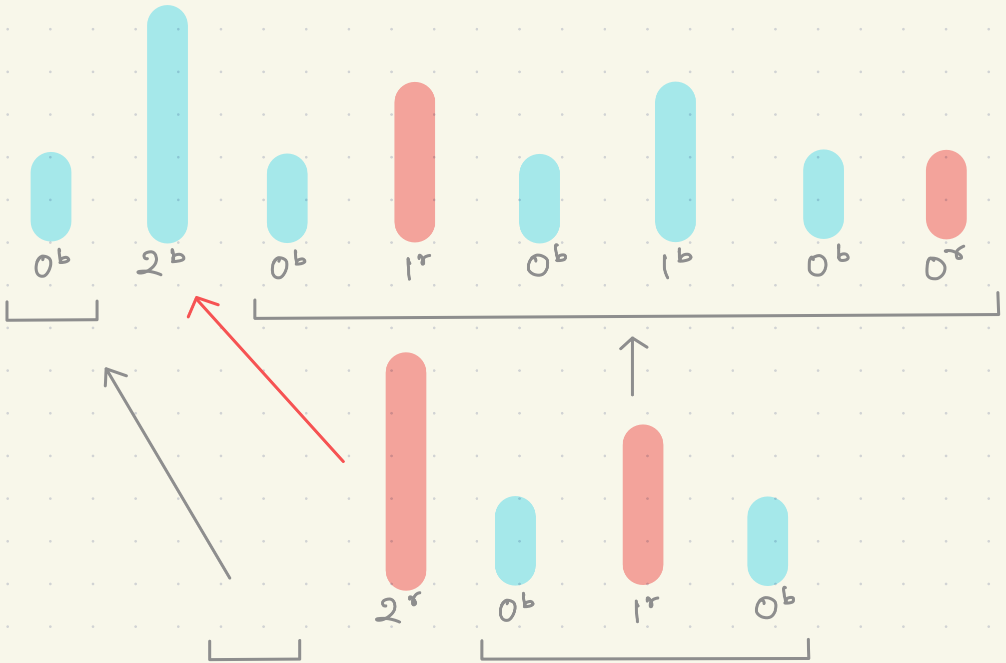
Block Order



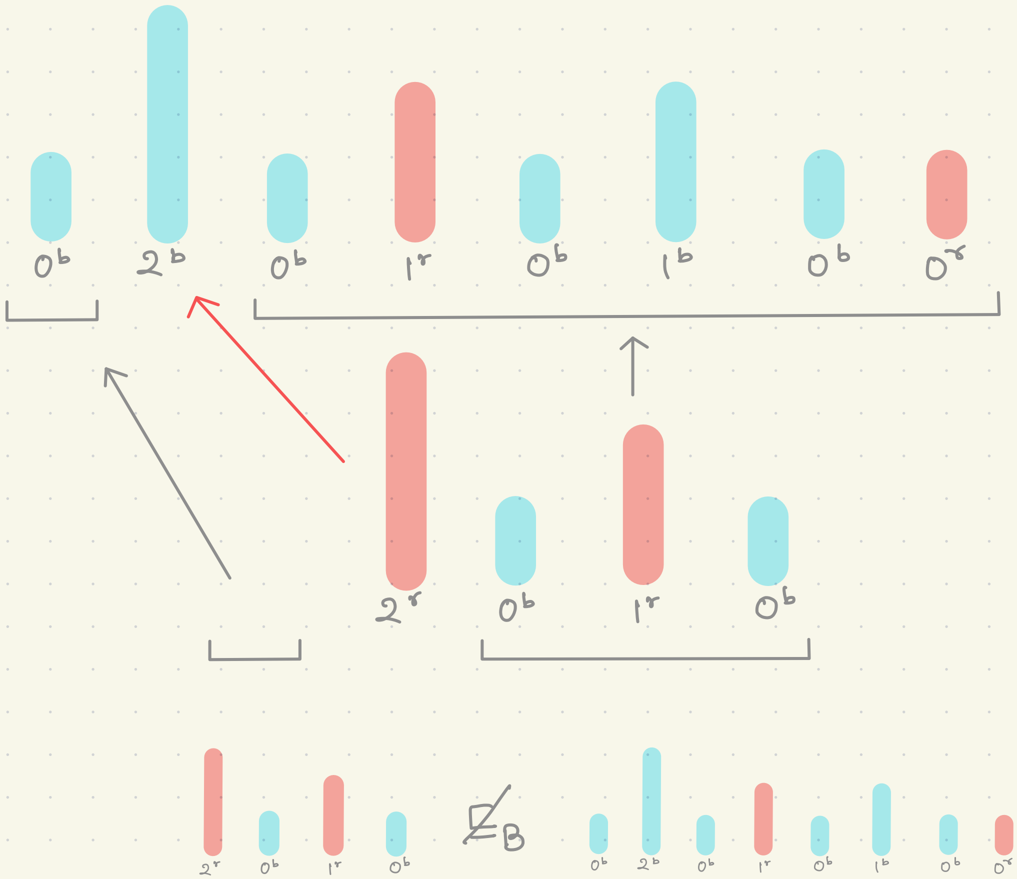
Block Order



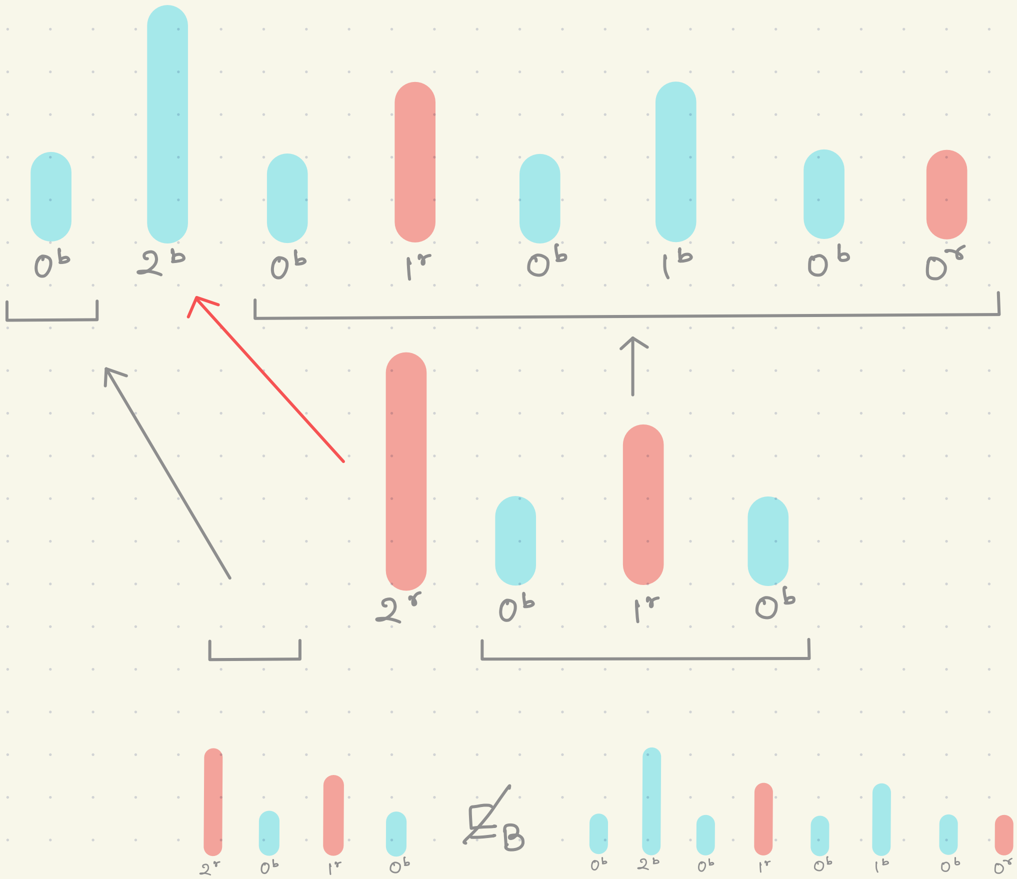
Block Order



Block Order

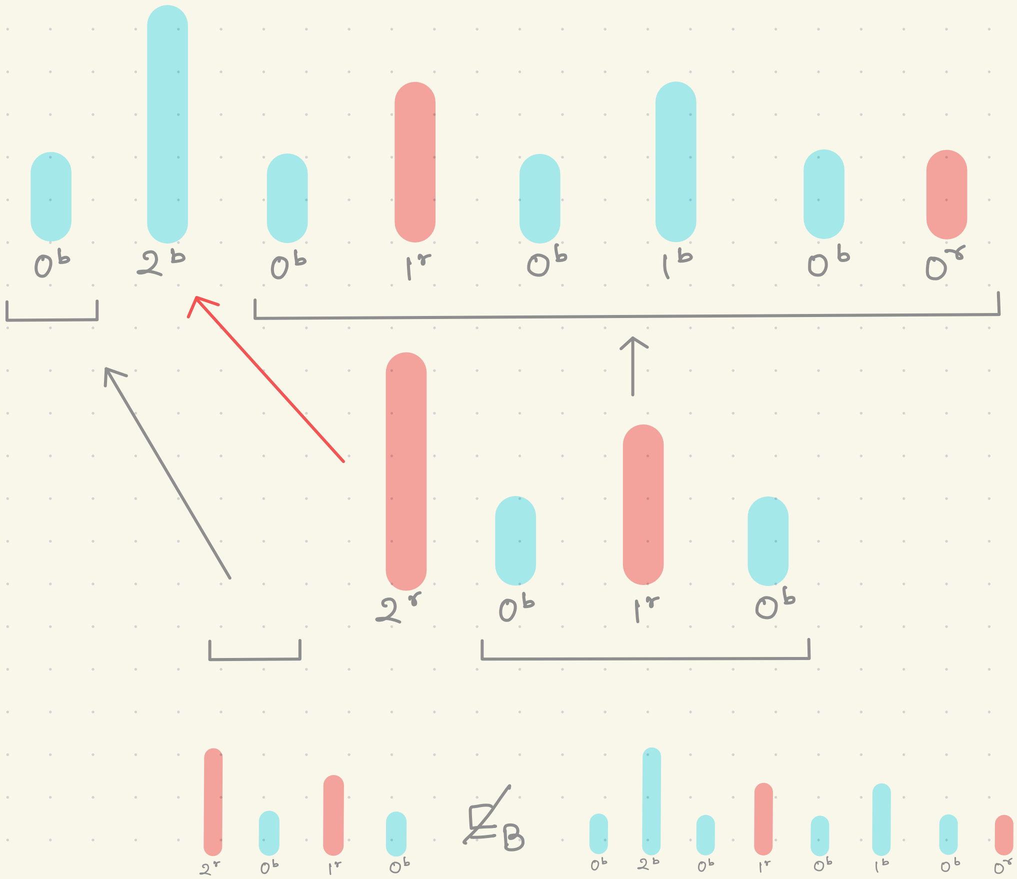


Block Order



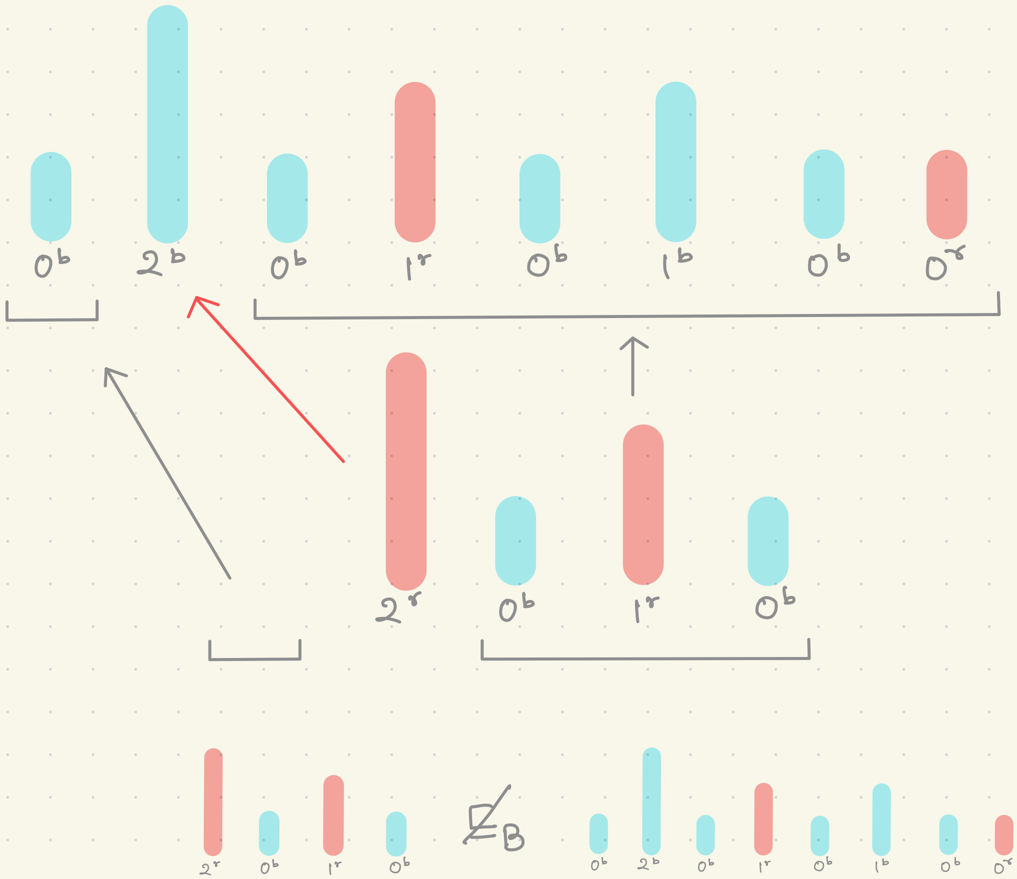
- Split word into blocks.

Block Order



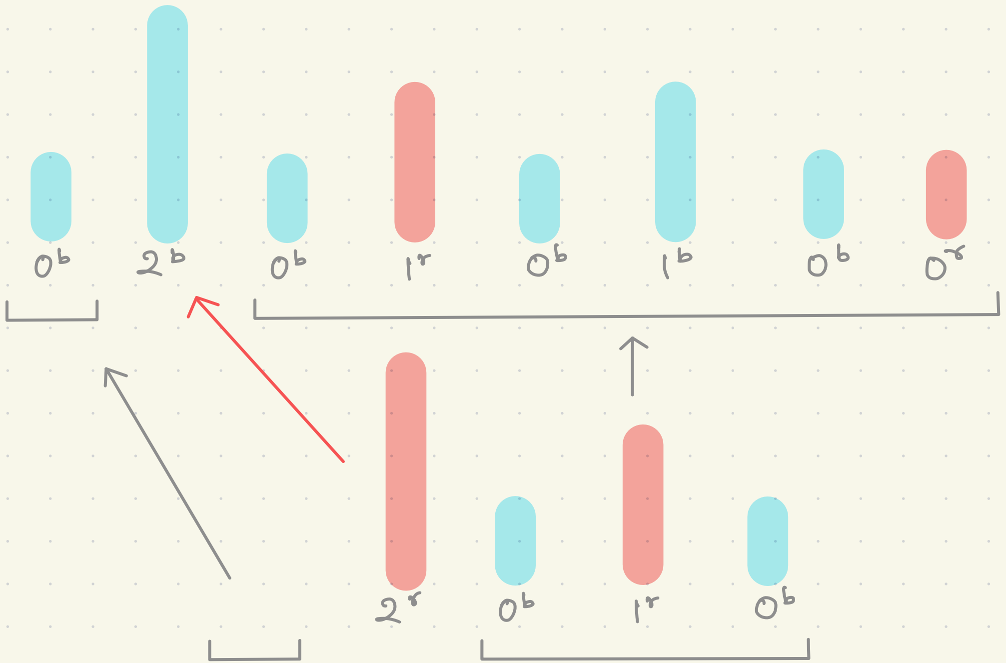
- Split word into blocks.
- Monotonic mapping of recursively embeddable blocks.

Block Order



- Split word into blocks.
- Monotonic mapping of recursively embeddable blocks.
- Highest priority letters occur in appropriate positions.

Block Order



- Split word into blocks.
- Monotonic mapping of recursively embeddable blocks.
- Highest priority letters occur in appropriate positions.

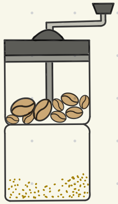
[In this talk] One letter per priority.

Block Order

Considers priority over letters

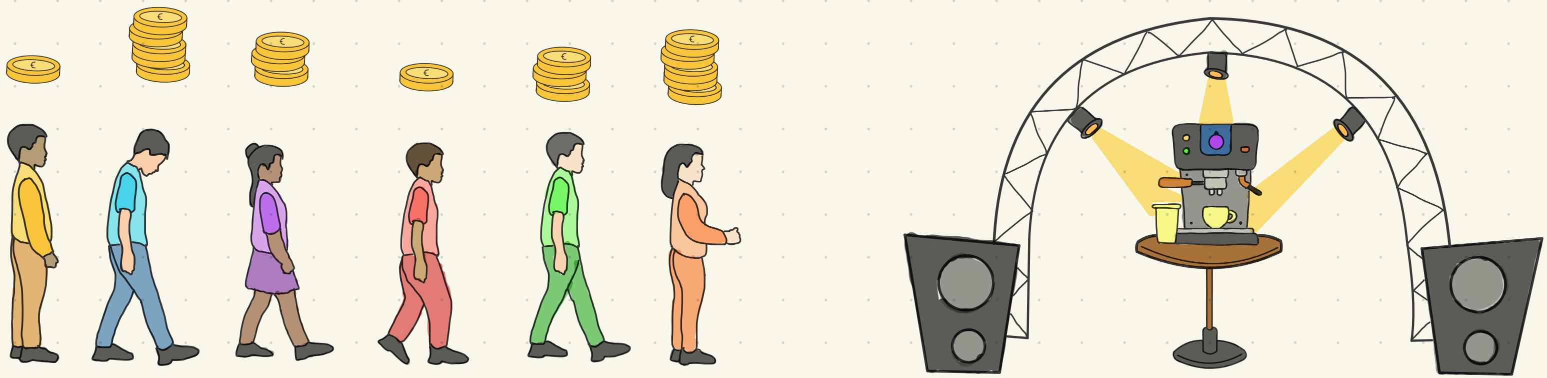
Block Order

Considers priority over letters

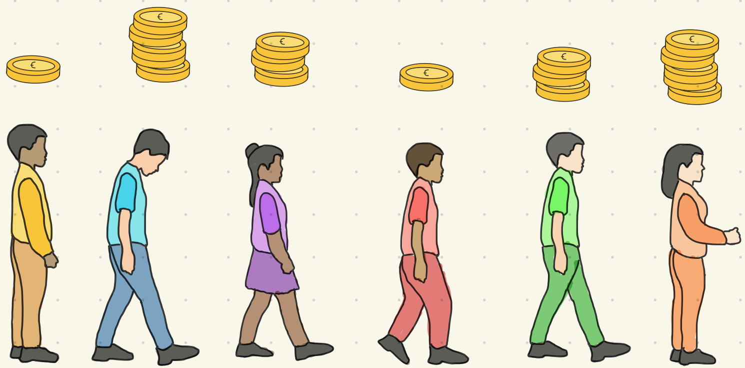


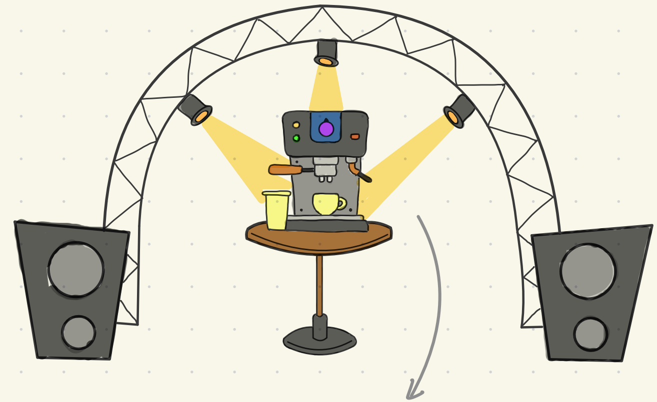
Refines Priortised Superseding Order [Haase et. al., 2014] and Subword Order

Block Downward Closures



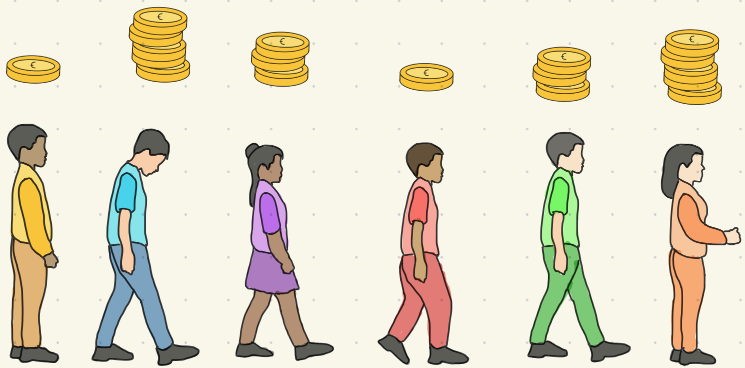
Block Downward Closures

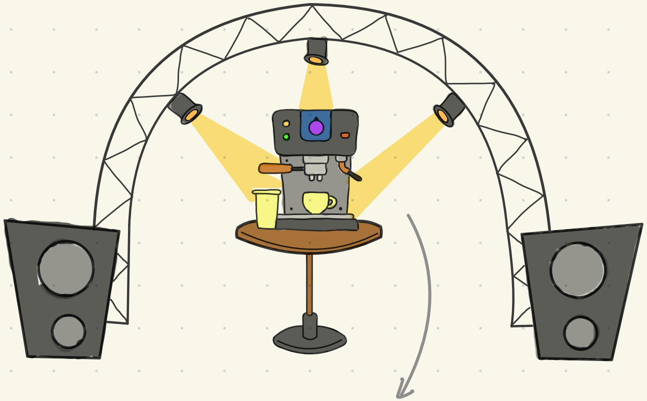




Sees only a block smaller queue of original queue.

Block Downward Closures

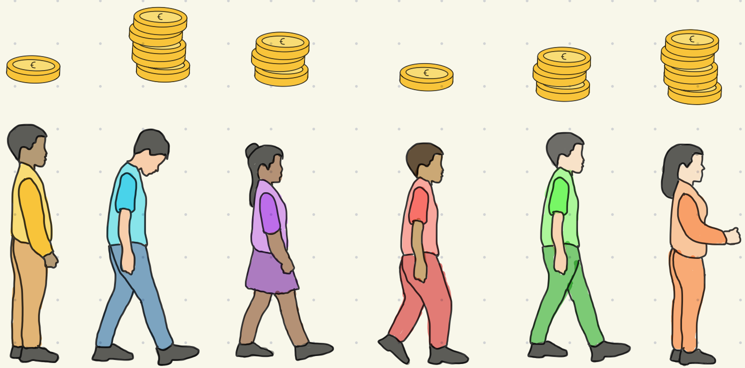


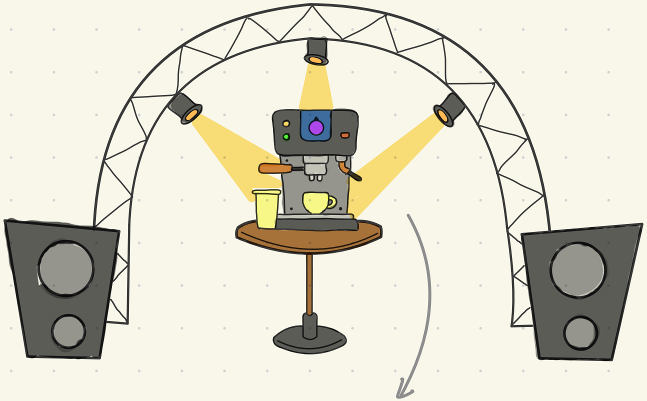


Sees only a block smaller queue of original queue.

Overapproximate to consider all block smaller behaviors shown by original machine.

Block Downward Closures



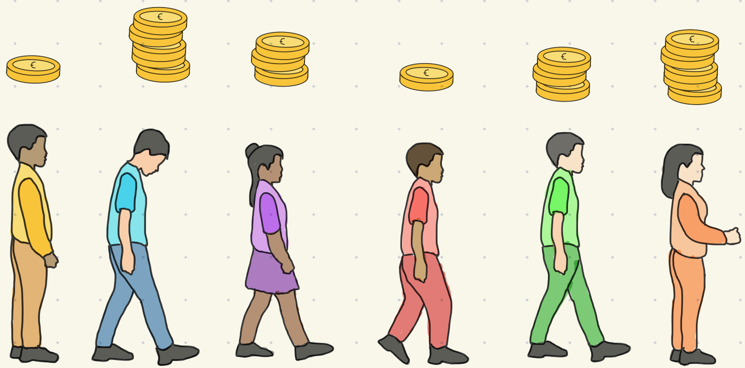


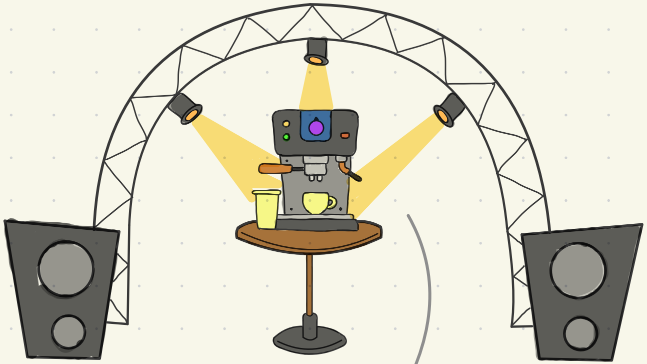
Downward closure of the machine.

Sees only a block smaller queue of original queue.

Overapproximate to consider all block smaller behaviors shown by original machine.

Block Downward Closures





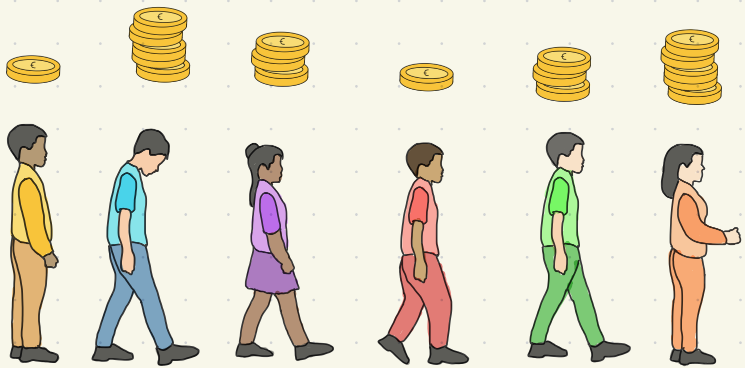
Downward closure of the machine.

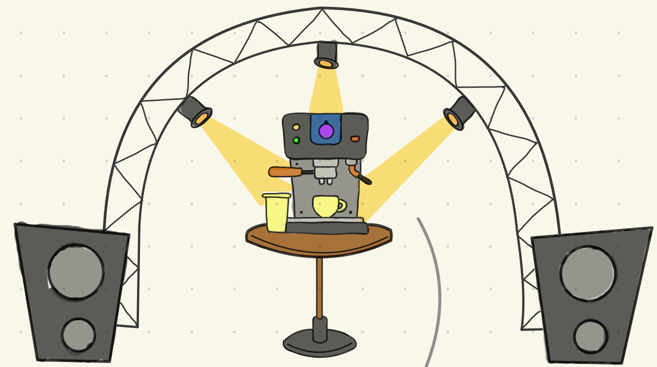
$$L\Downarrow = \{ w \mid v \in L,\ w \sqsubseteq_B v \}$$

Sees only a block smaller queue of original queue.

Overapproximate to consider all block smaller behaviors shown by original machine.

Block Downward Closures





Downward closure of the machine.

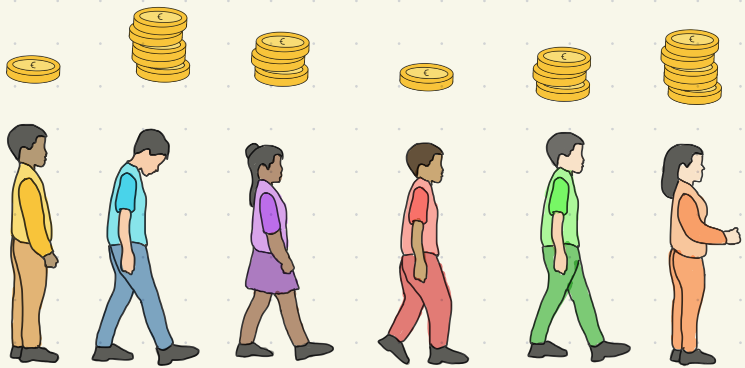
$L\Downarrow = \{ w \mid v \in L,\ w \sqsubseteq_B v \}$

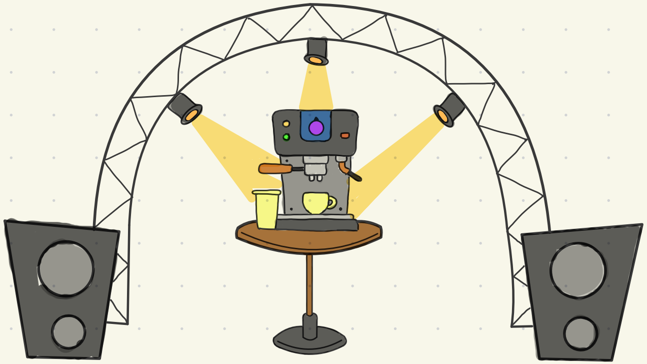
Always accepted by finite state machine.

Sees only a block smaller queue of original queue.

Overapproximate to consider all block smaller behaviors shown by original machine.

Block Downward Closures





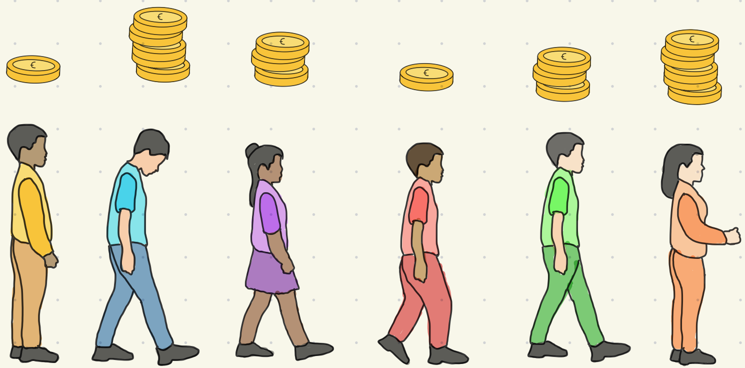
Downward closure of the machine.

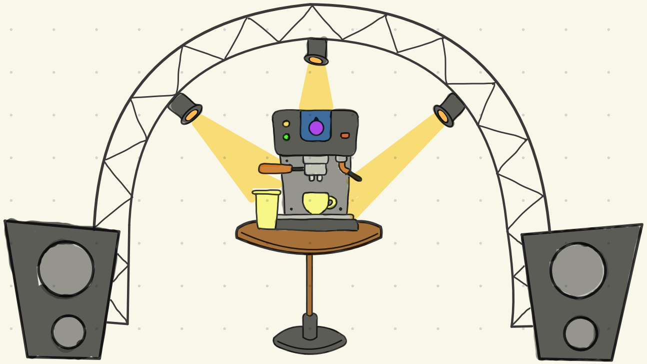
Might not be computable!

$L{\Downarrow} = \{ w \mid v \in L,\ w \sqsubseteq_B v \}$

Always accepted by finite state machine.

Block Downward Closures





Downward closure of the machine.

$L\Downarrow = \{ w \mid v \in L,\ w \sqsubseteq_B v \}$

Always accepted by finite state machine.

Might not be computable!

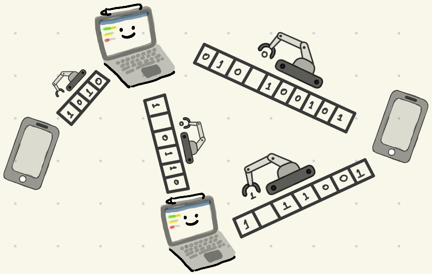
[This talk] Construction of such machine for pushdown machines.

Block Downward Closures

Coffee machine
at a concert

Block Downward Closures

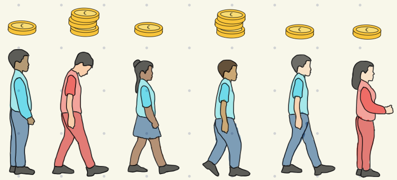
Coffee machine
at a concert



Network of systems
with congestion controlled
communication channels

Overview

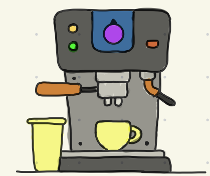
Block Order



- Considers priorities
- Refines subword order and PSO

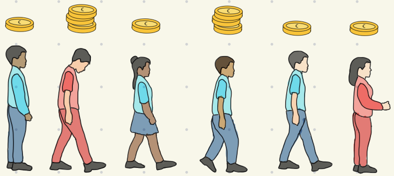
Simple Machines

Pushdown Machines



Overview

Block Order



- Considers priorities
- Refines subword order and PSO

Simple Machines

Pushdown Machines

Finite State Machines

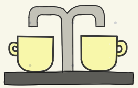
Finite state machine

accepting a language L

Finite State Machines

Finite state machine

accepting a language L



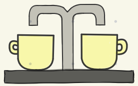
A transducer

takes an input word, outputs another word.

Finite State Machines

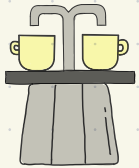
Finite state machine

accepting a language L



A transducer

takes an input word, outputs another word.



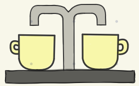
Another finite state machine

accepting block downward closure $L\Downarrow$

Finite State Machines

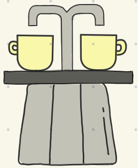
Finite state machine

accepting a language L



A transducer

takes an input word, outputs another word.



Another finite state machine

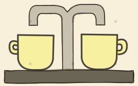
accepting block downward closure $L\Downarrow$

Construct a transducer that outputs block smaller words for a given word.

Finite State Machines

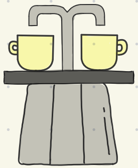
Finite state machine

accepting a language L



A transducer

takes an input word, outputs another word.

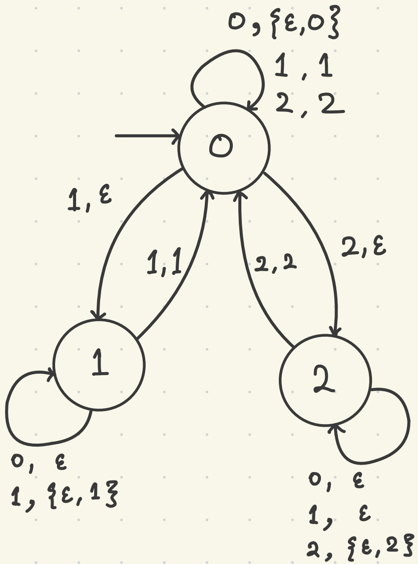


Another finite state machine

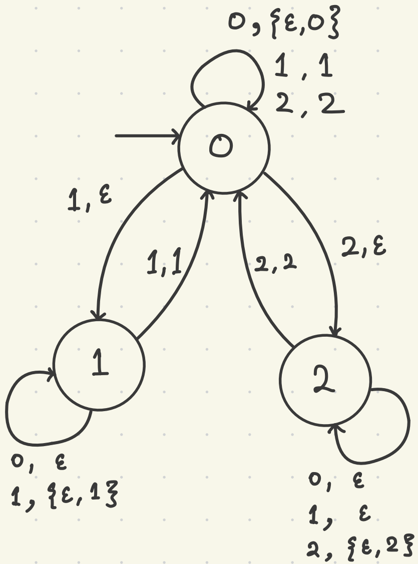
accepting block downward closure $L\Downarrow$

Construct a transducer that outputs block smaller words for a given word.

Transducer for Block Order

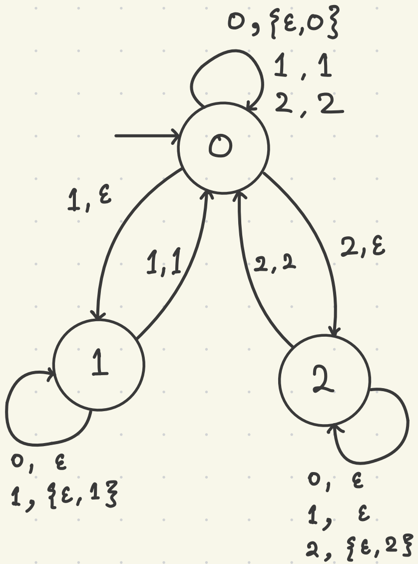


Transducer for Block Order

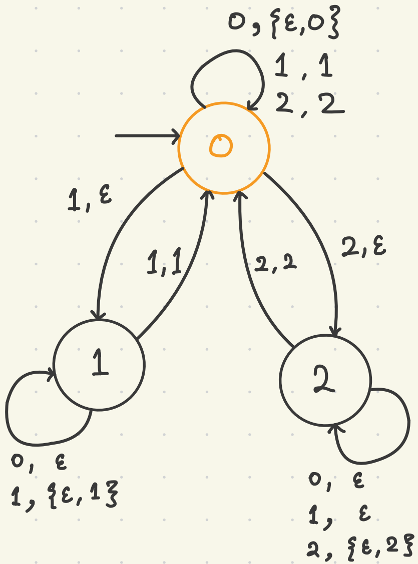


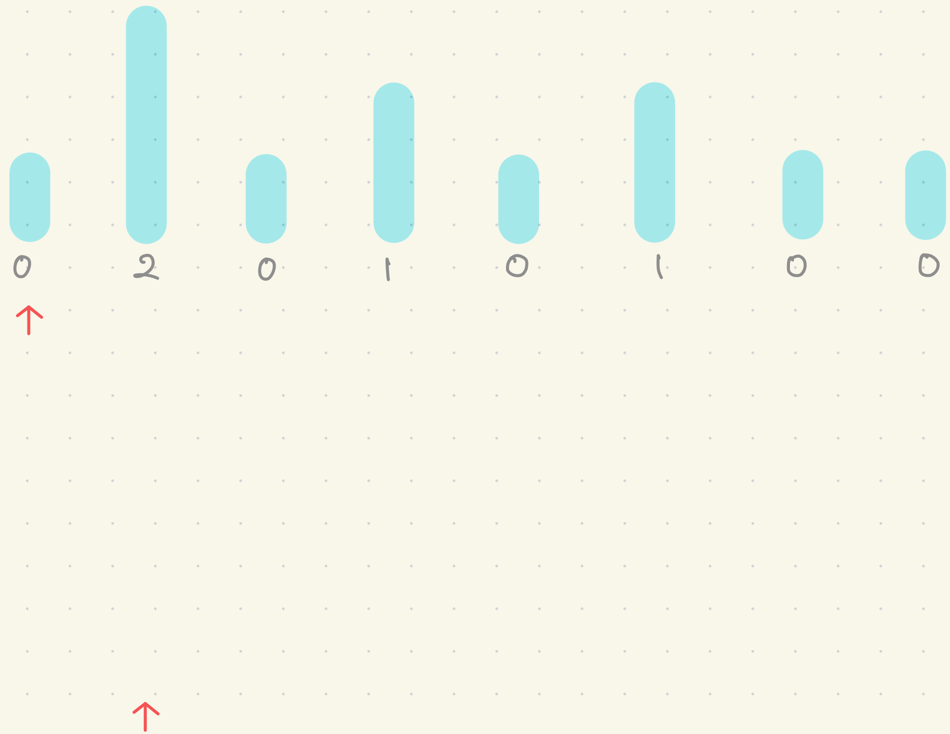
[Idea] If i is skipped then skip all $j < i$, until an i is output again.

Transducer for Block Order

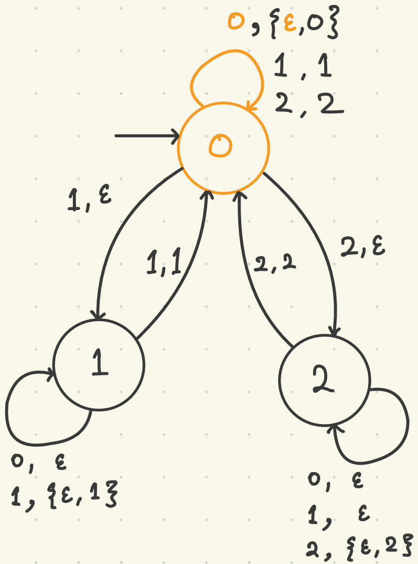


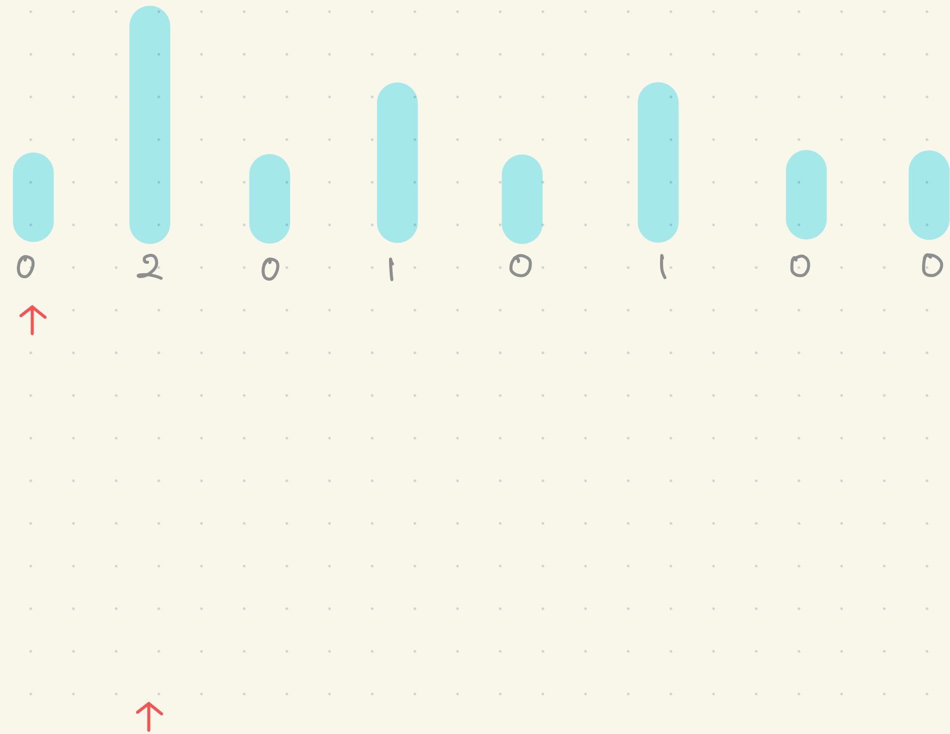
Transducer for Block Order



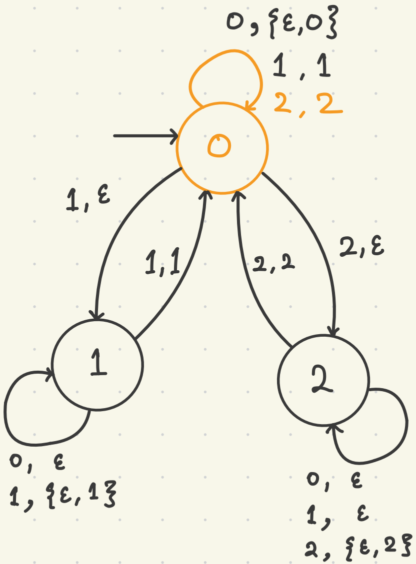


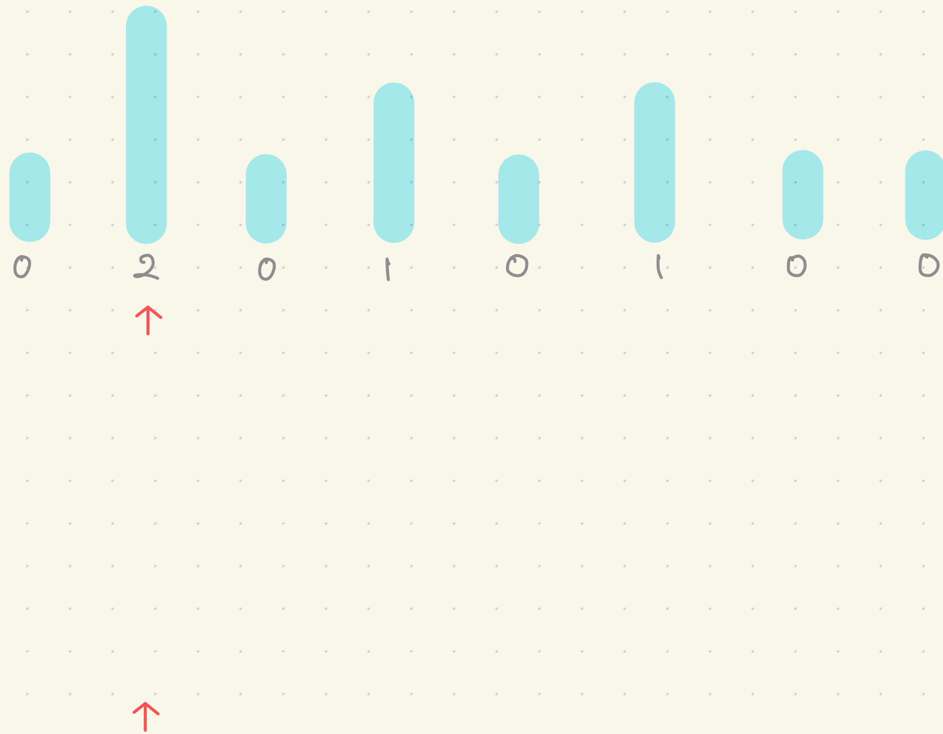
Transducer for Block Order



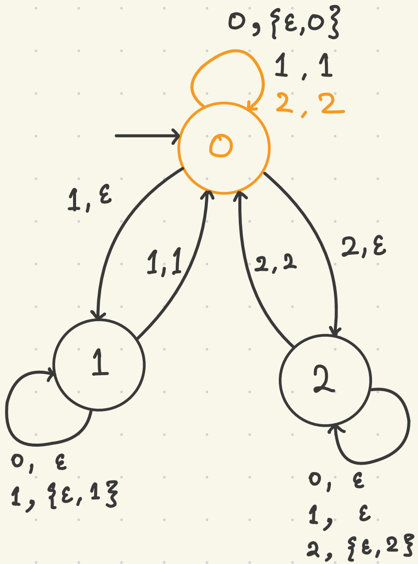


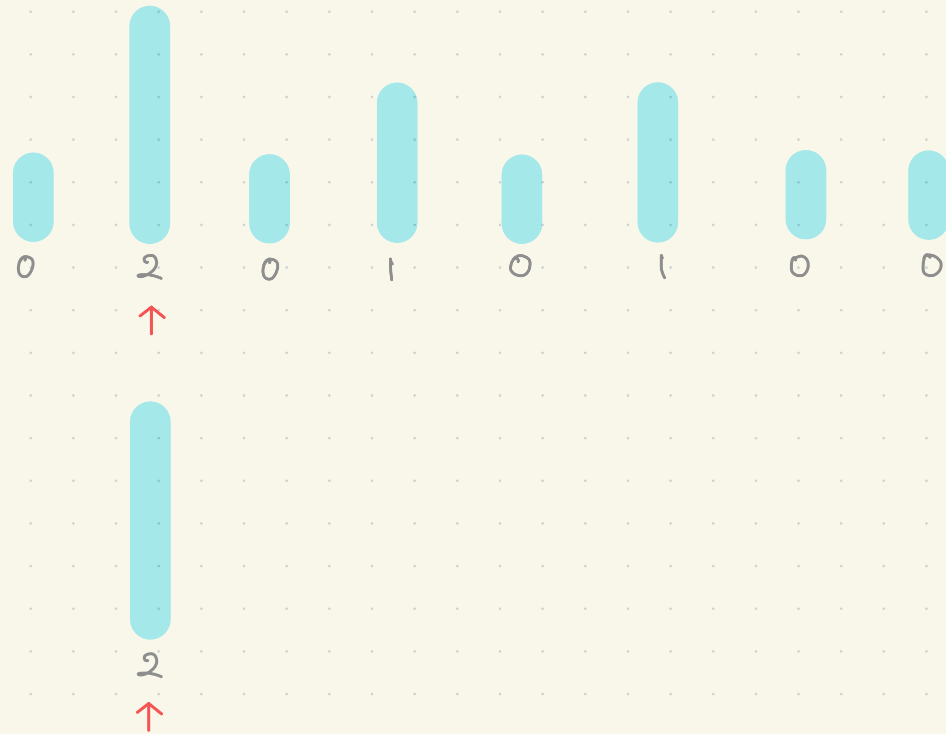
Transducer for Block Order



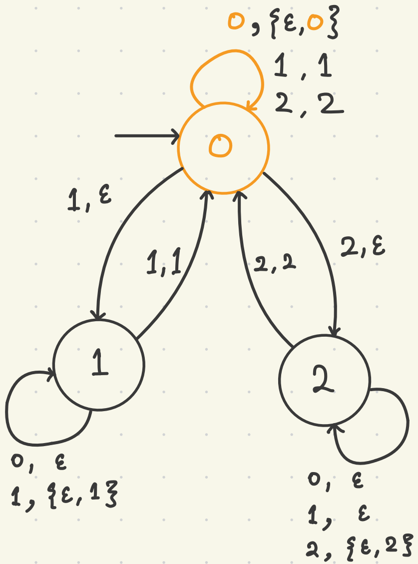


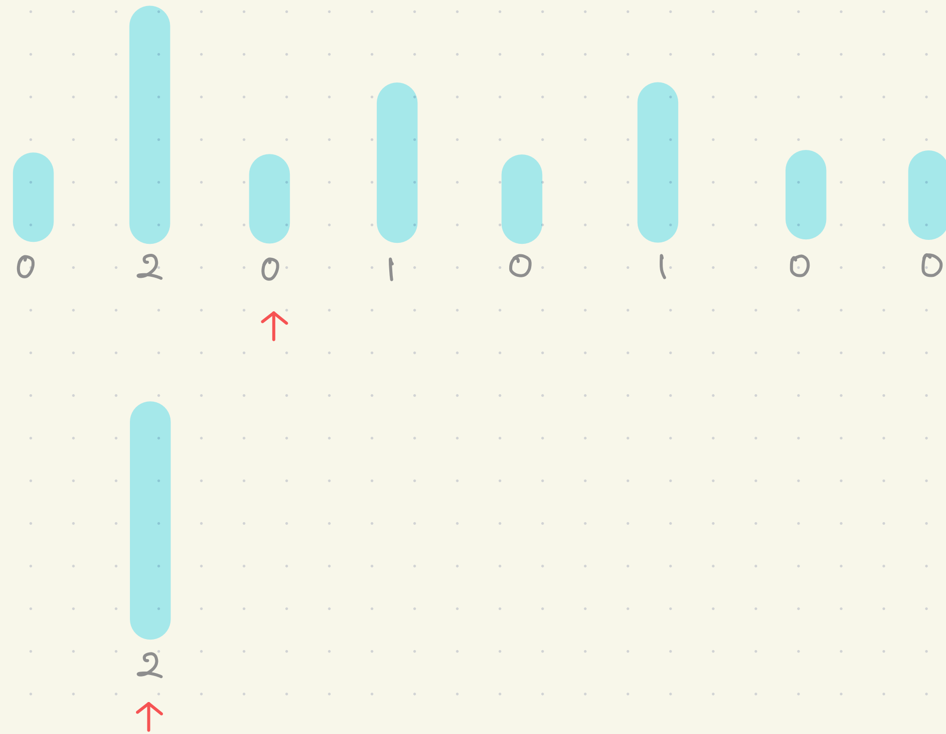
Transducer for Block Order



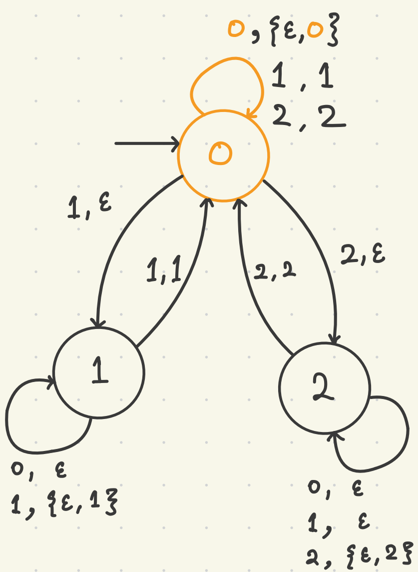


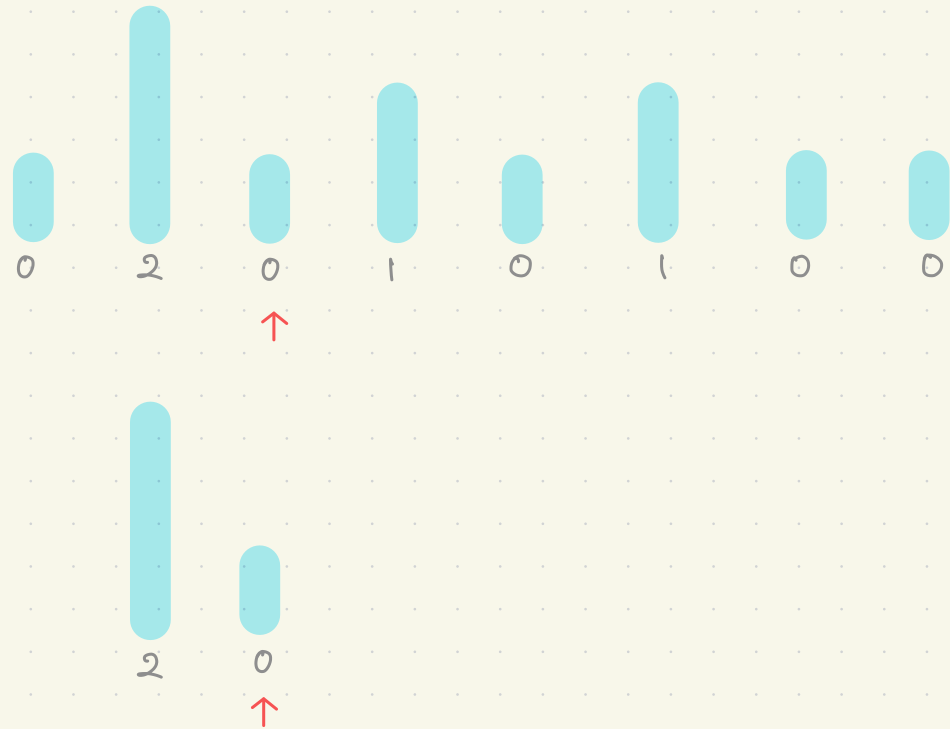
Transducer for Block Order



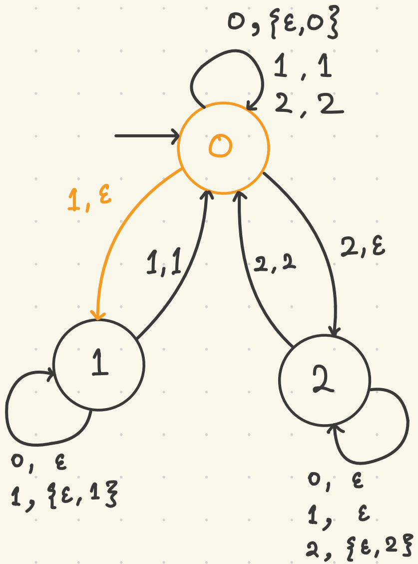


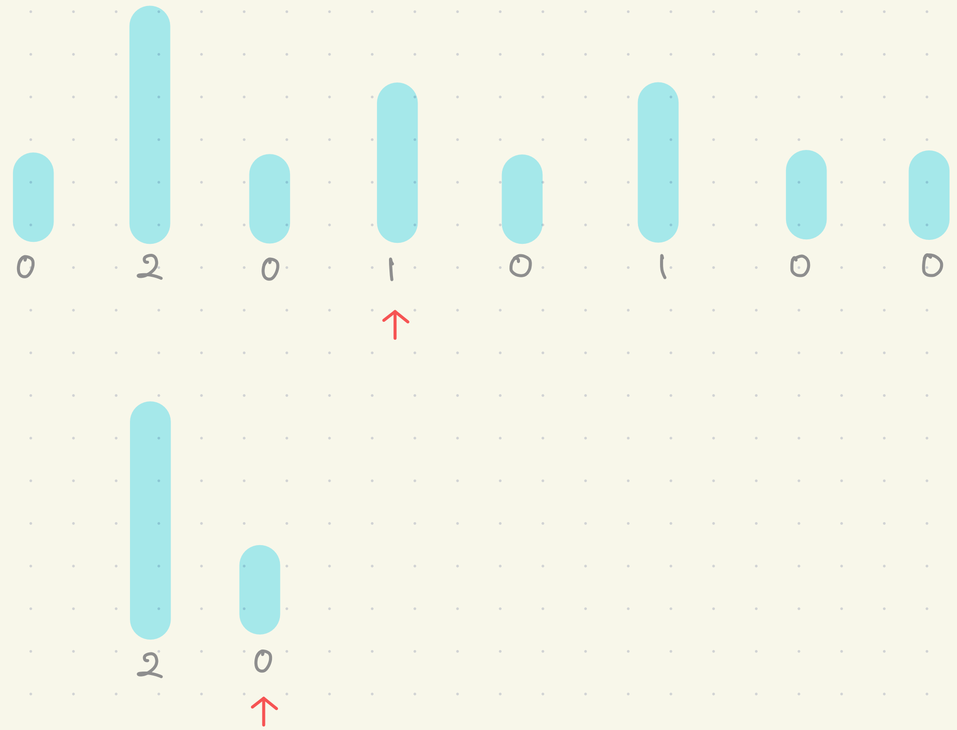
Transducer for Block Order



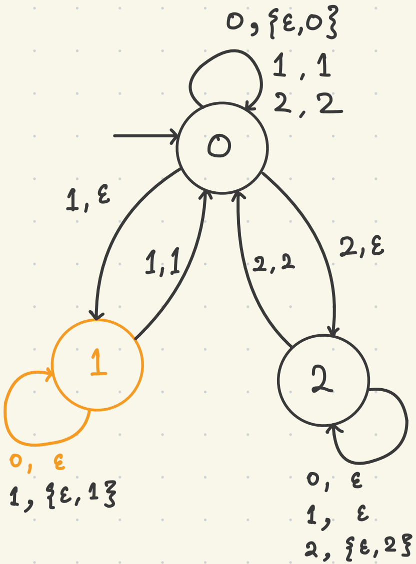


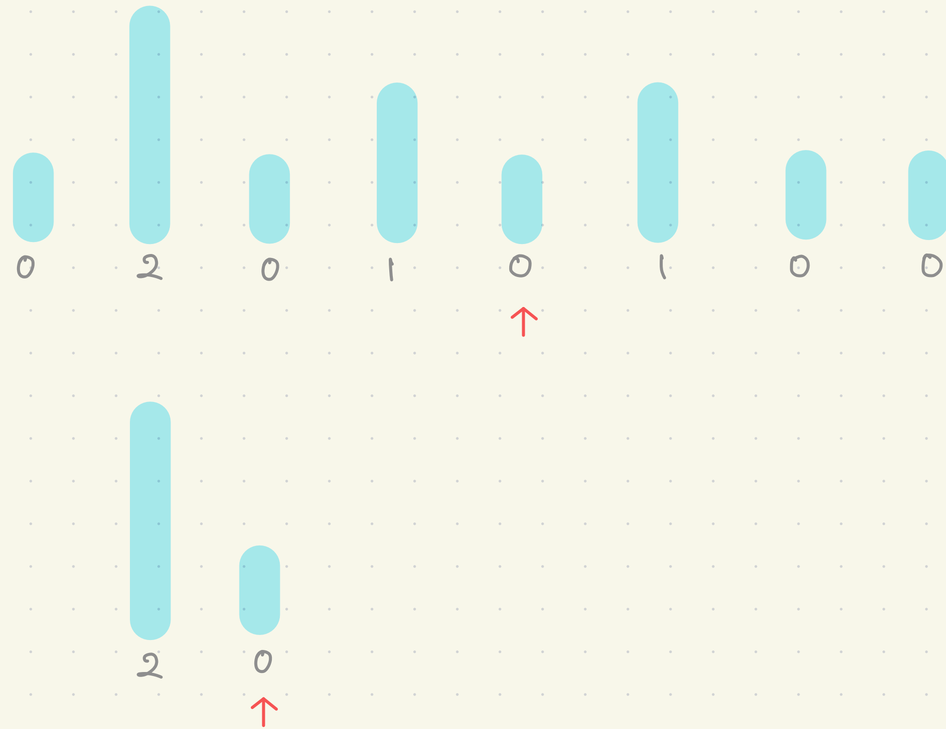
Transducer for Block Order



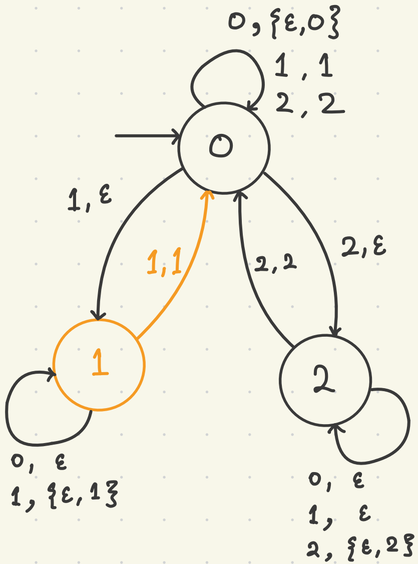


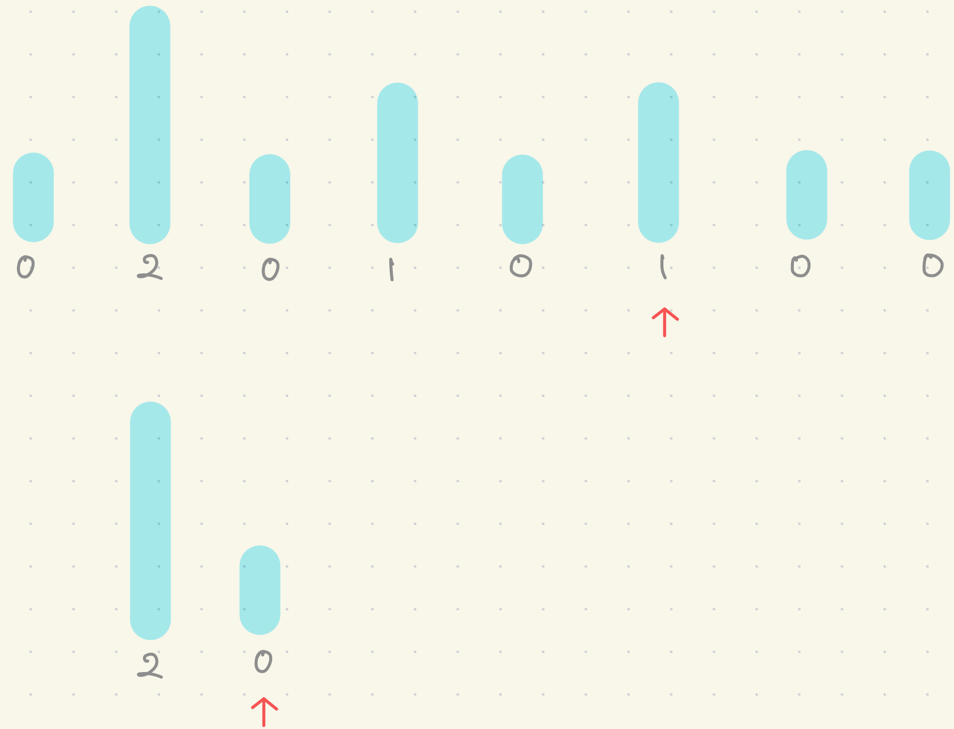
Transducer for Block Order



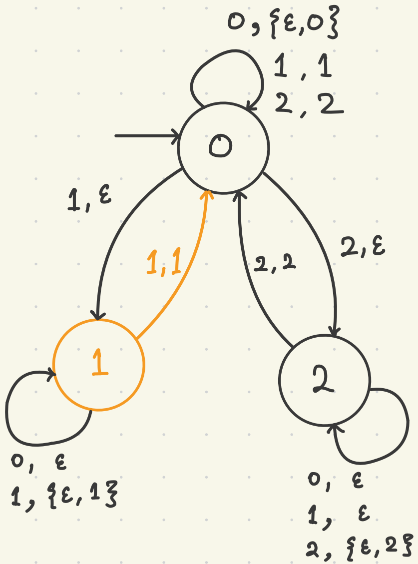


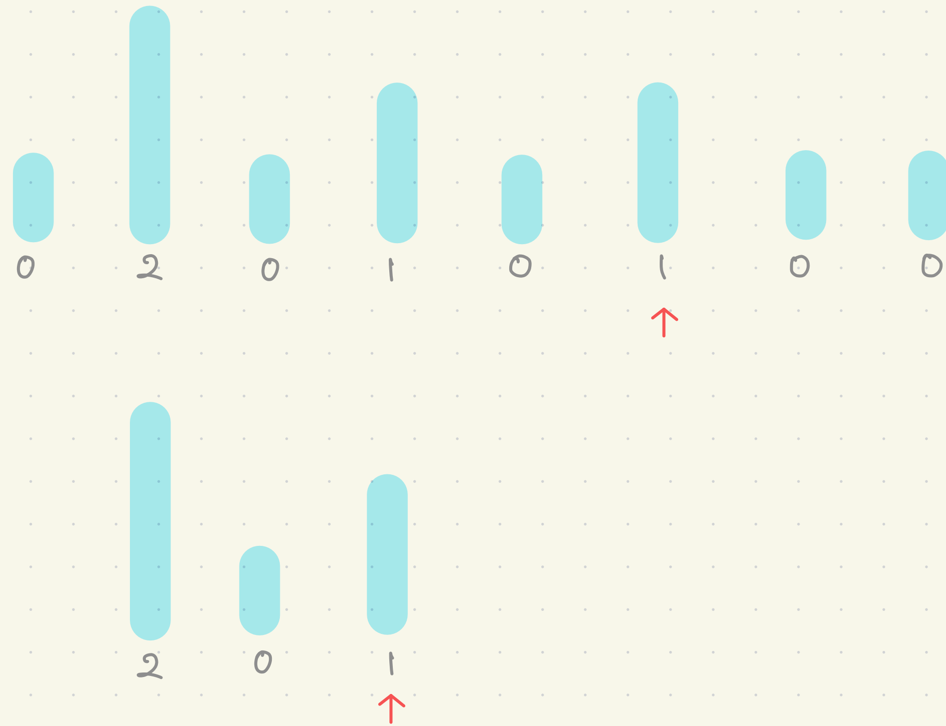
Transducer for Block Order



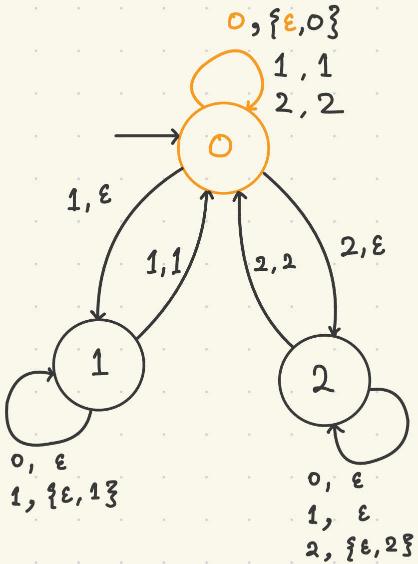


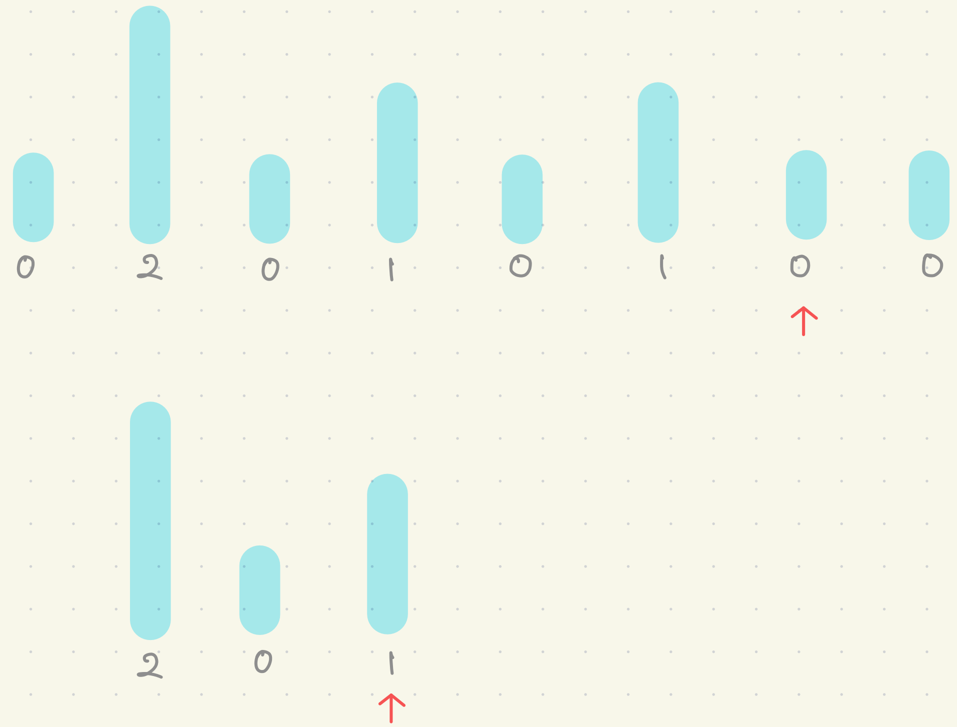
Transducer for Block Order



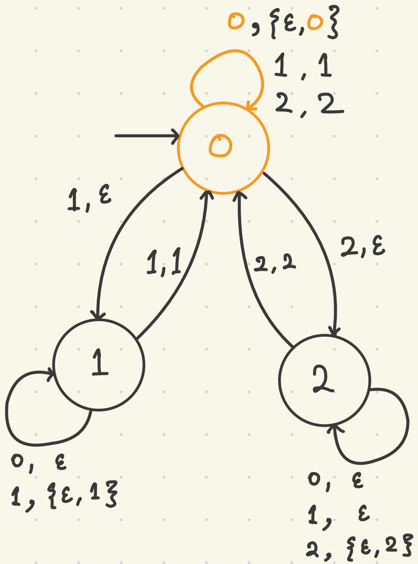


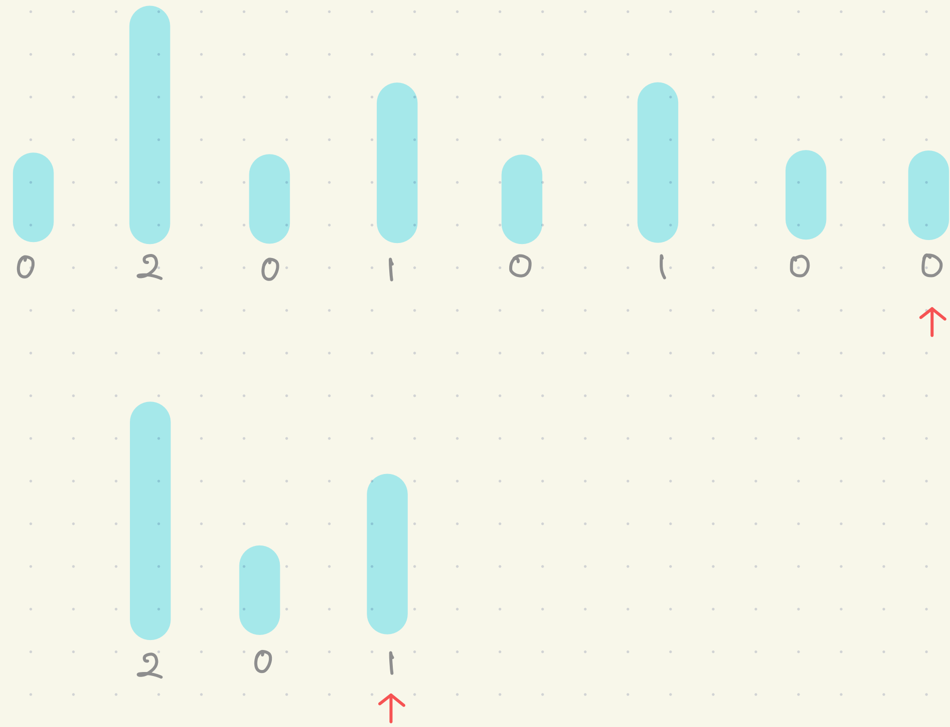
Transducer for Block Order



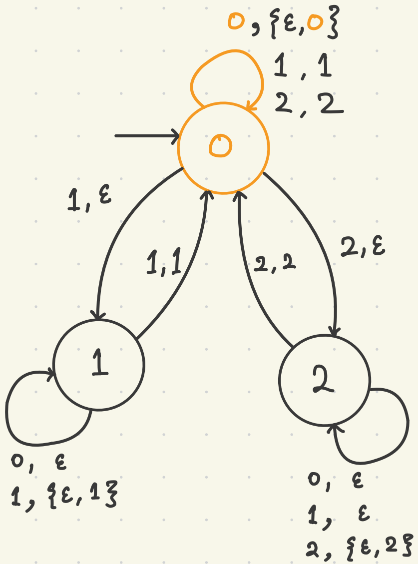


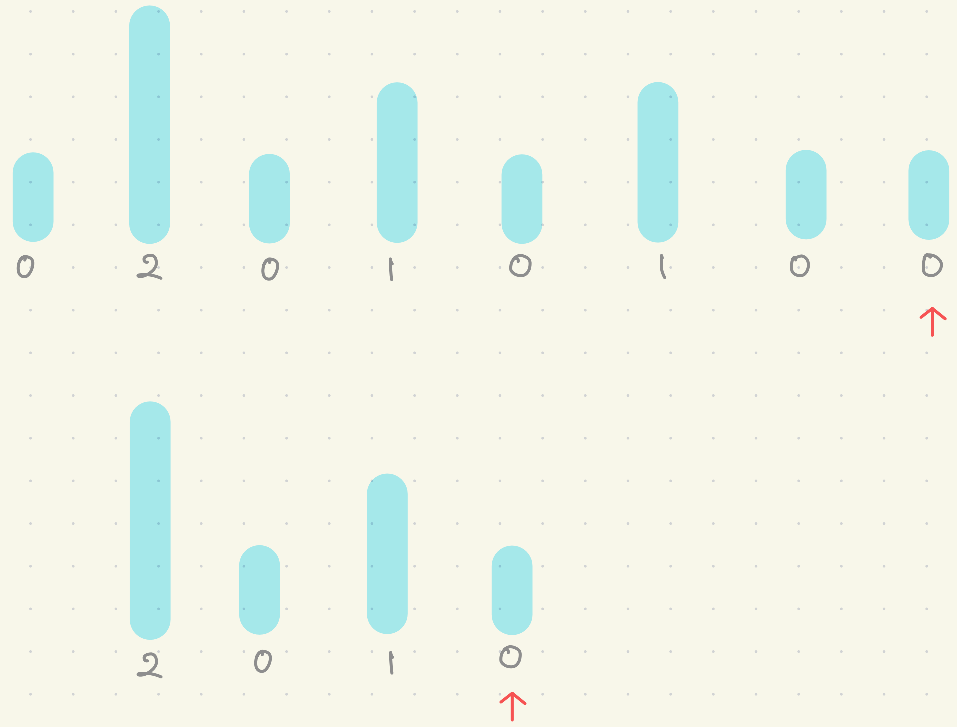
Transducer for Block Order



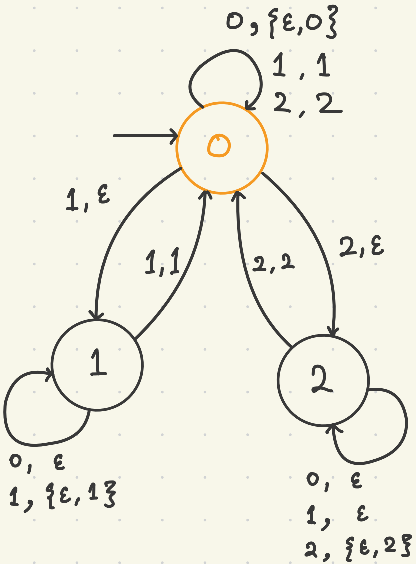


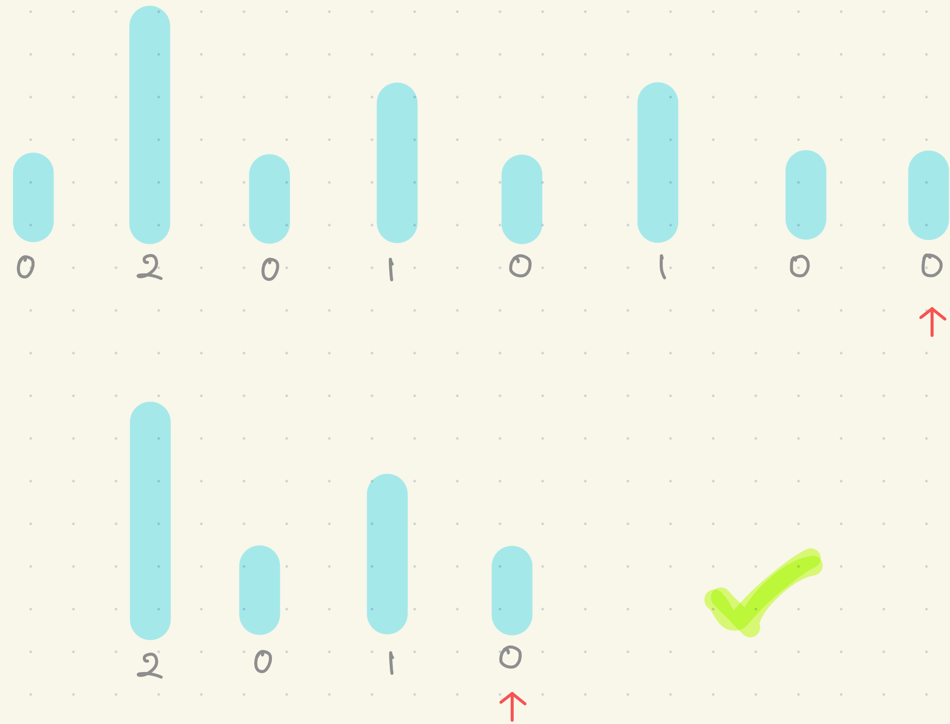
Transducer for Block Order



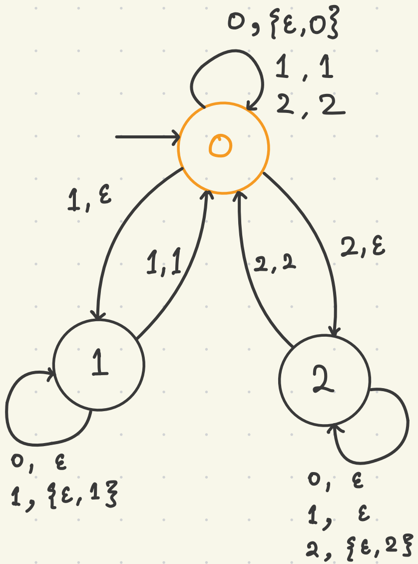


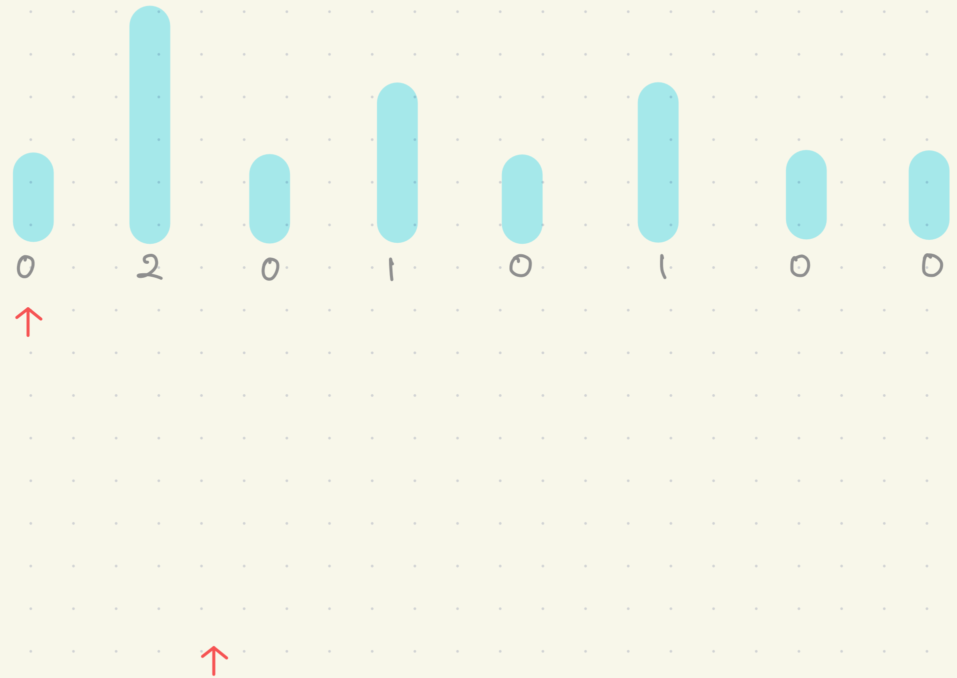
Transducer for Block Order



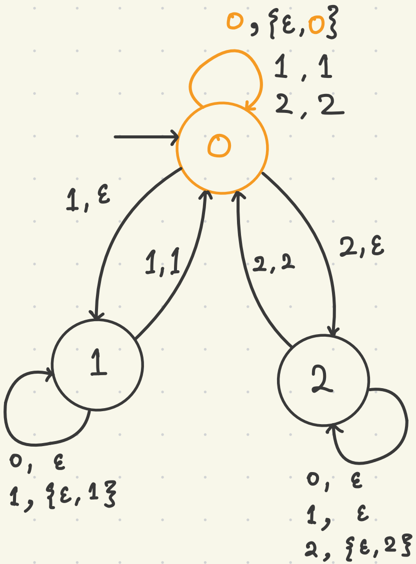


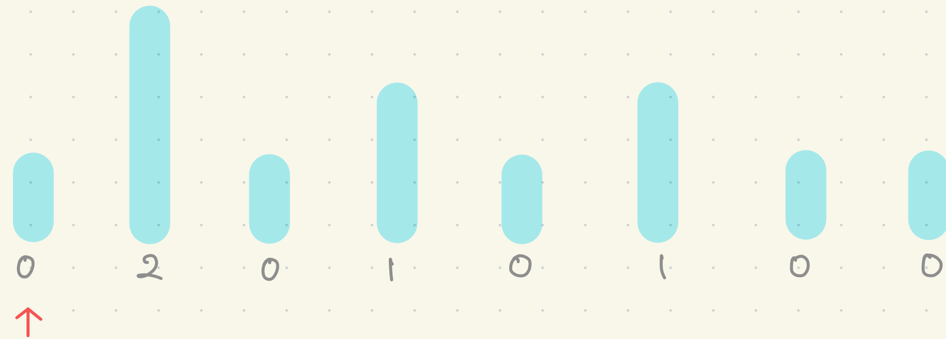
Transducer for Block Order



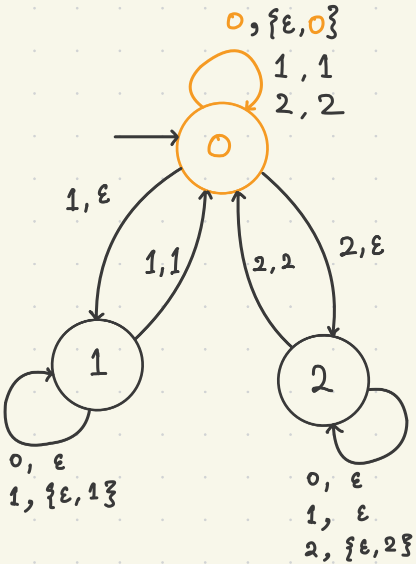


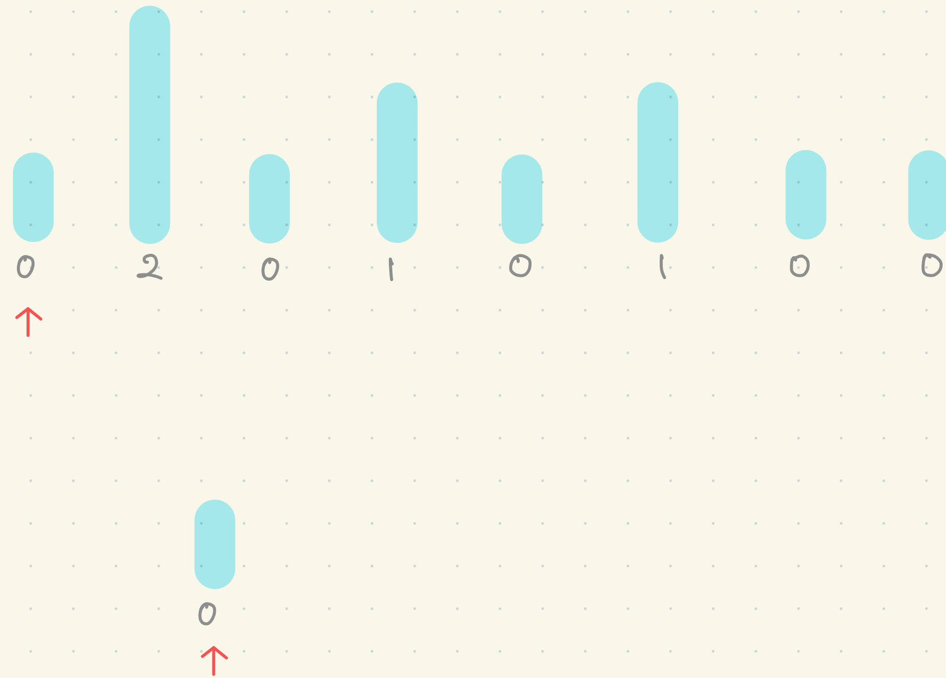
Transducer for Block Order



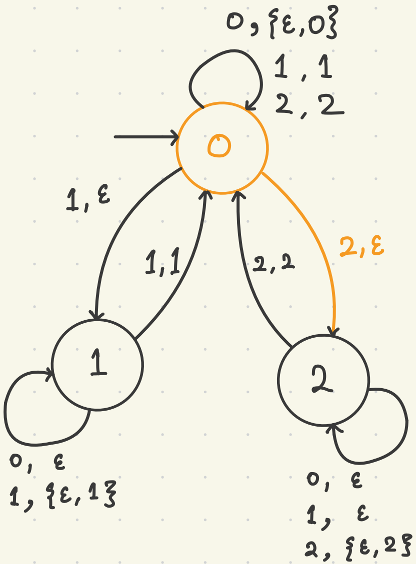


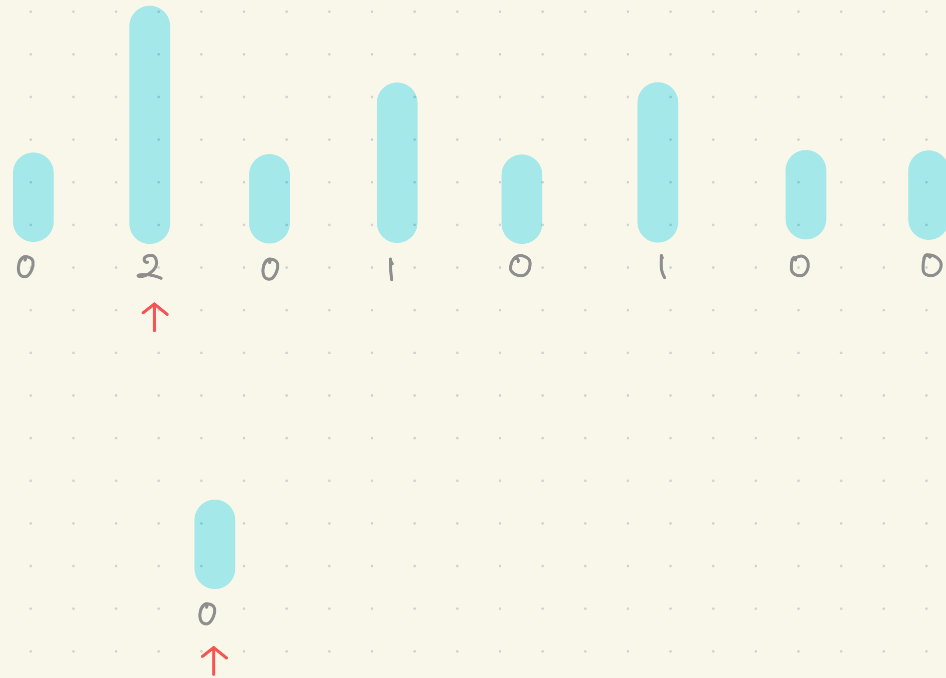
Transducer for Block Order



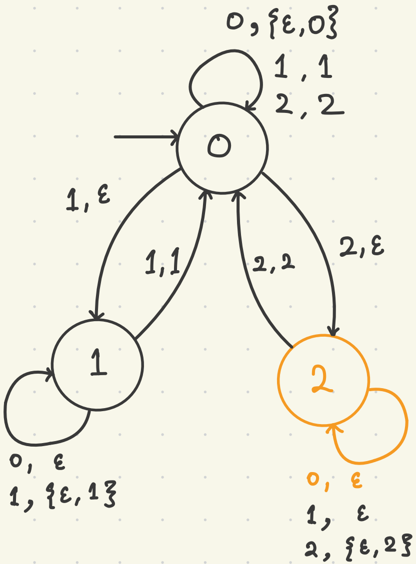


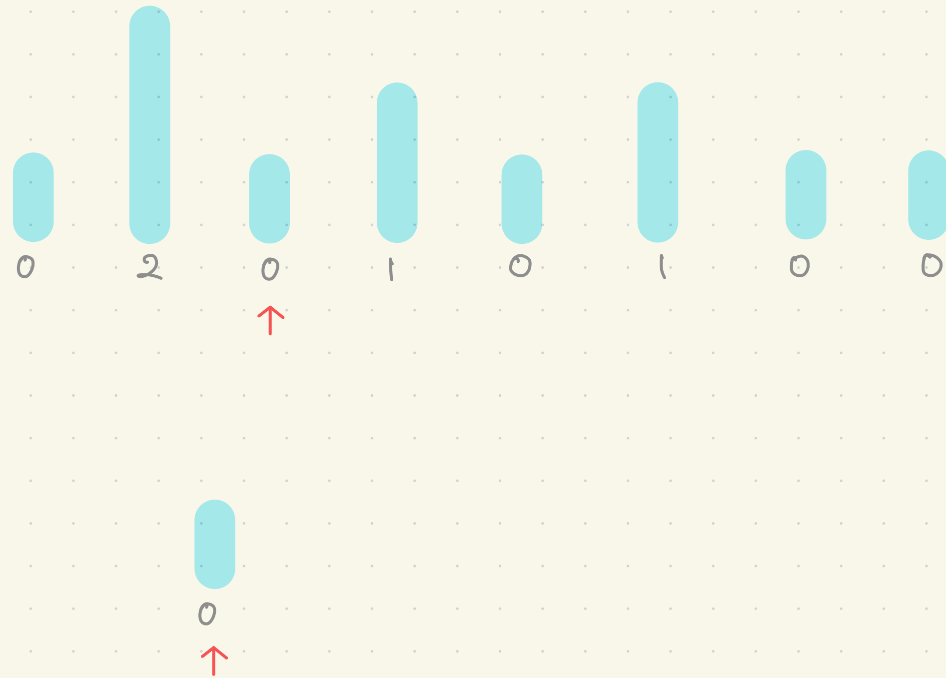
Transducer for Block Order



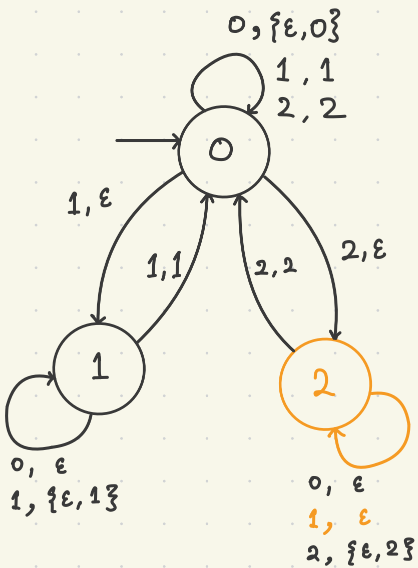


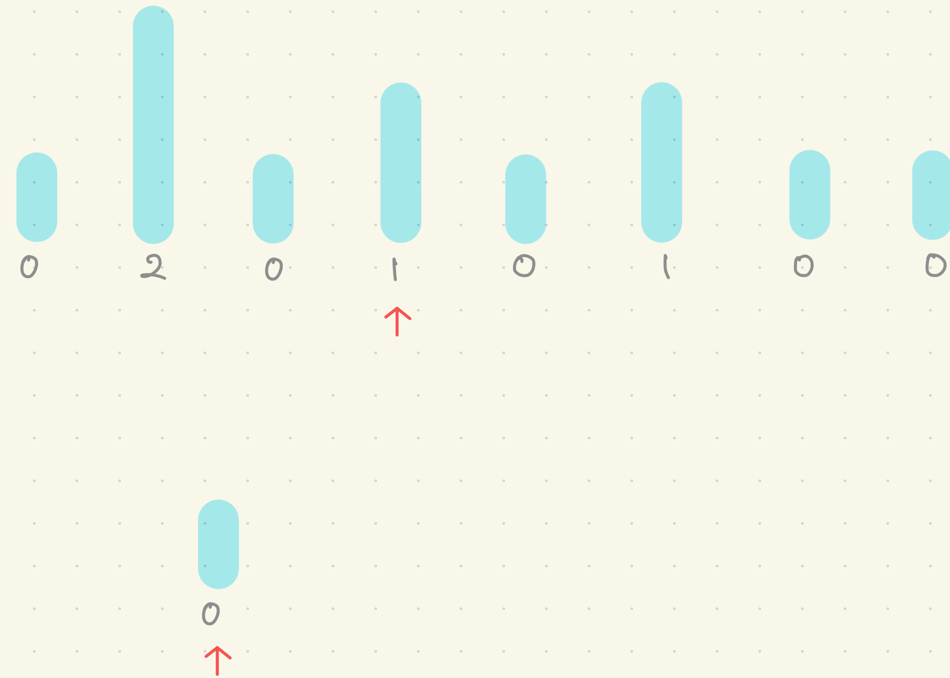
Transducer for Block Order



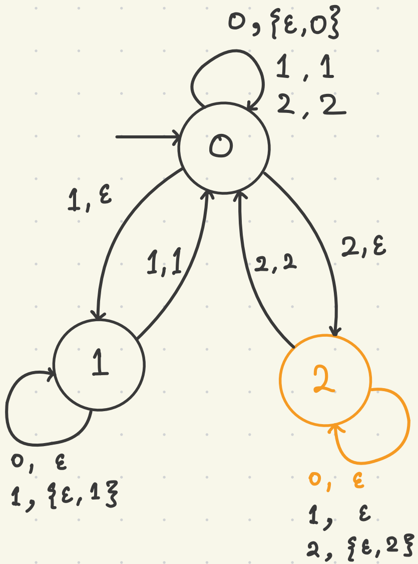


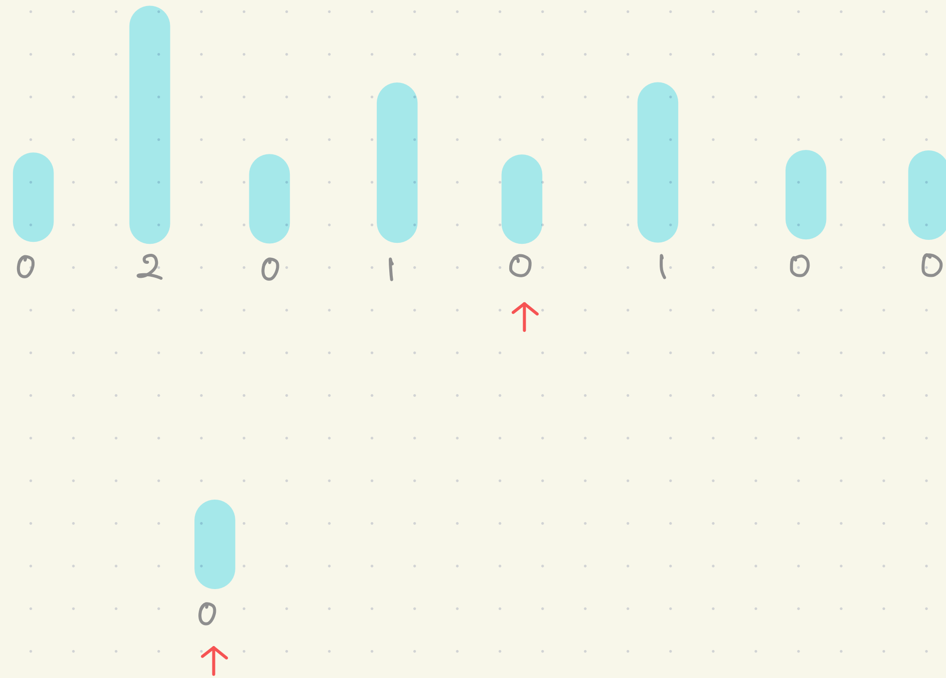
Transducer for Block Order



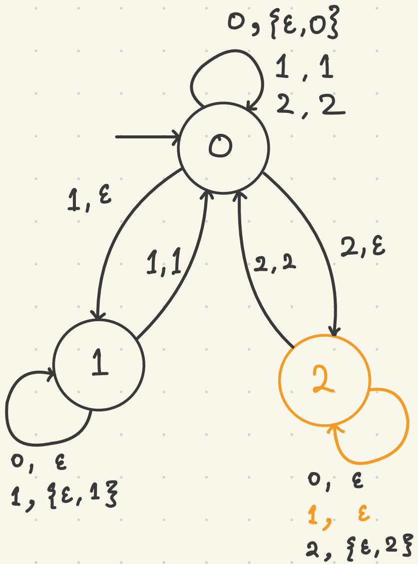


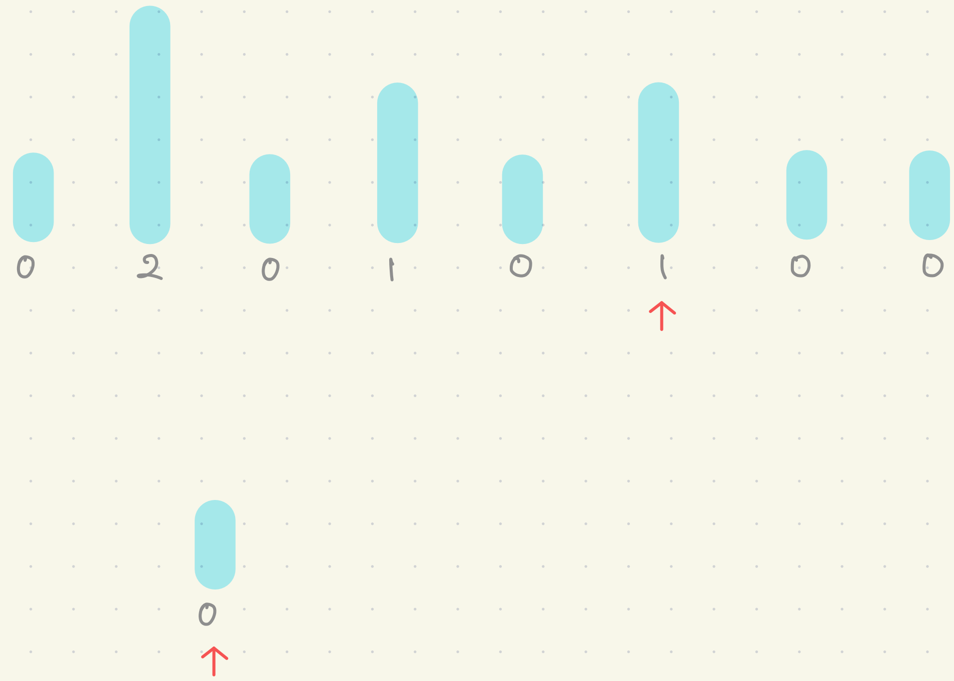
Transducer for Block Order



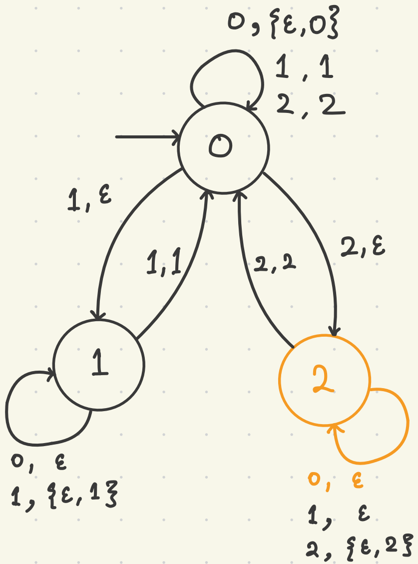


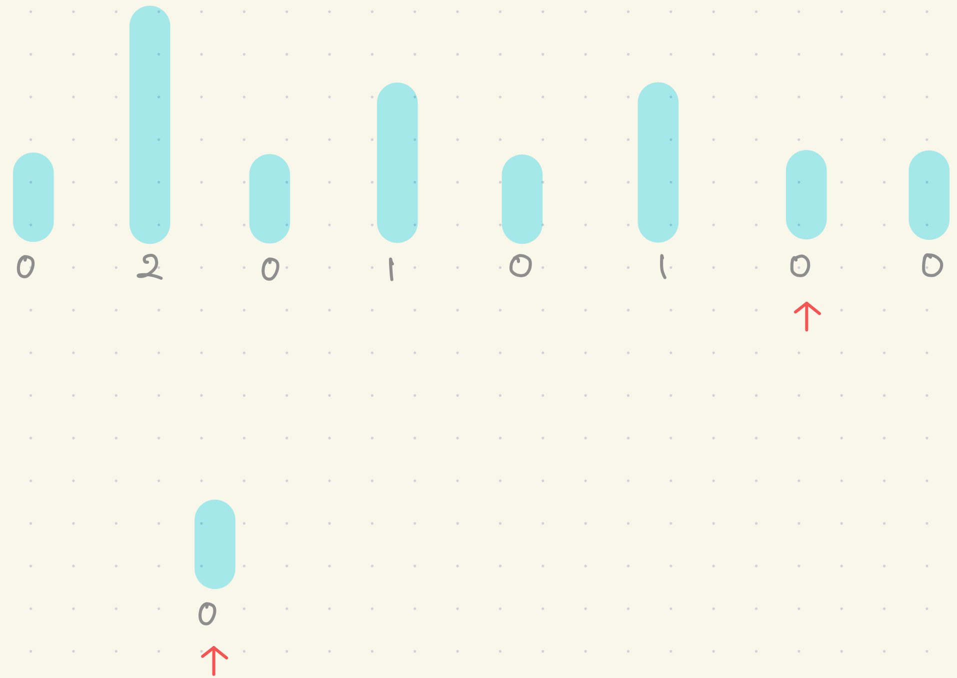
Transducer for Block Order



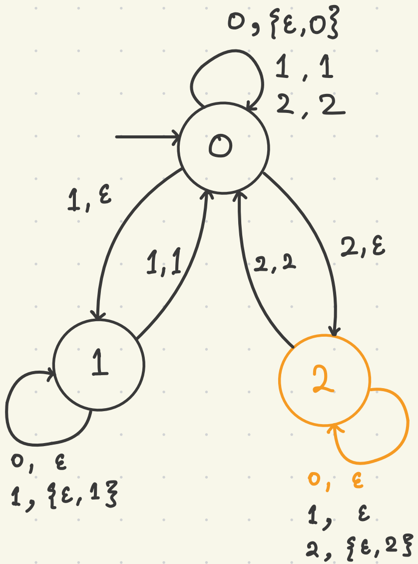


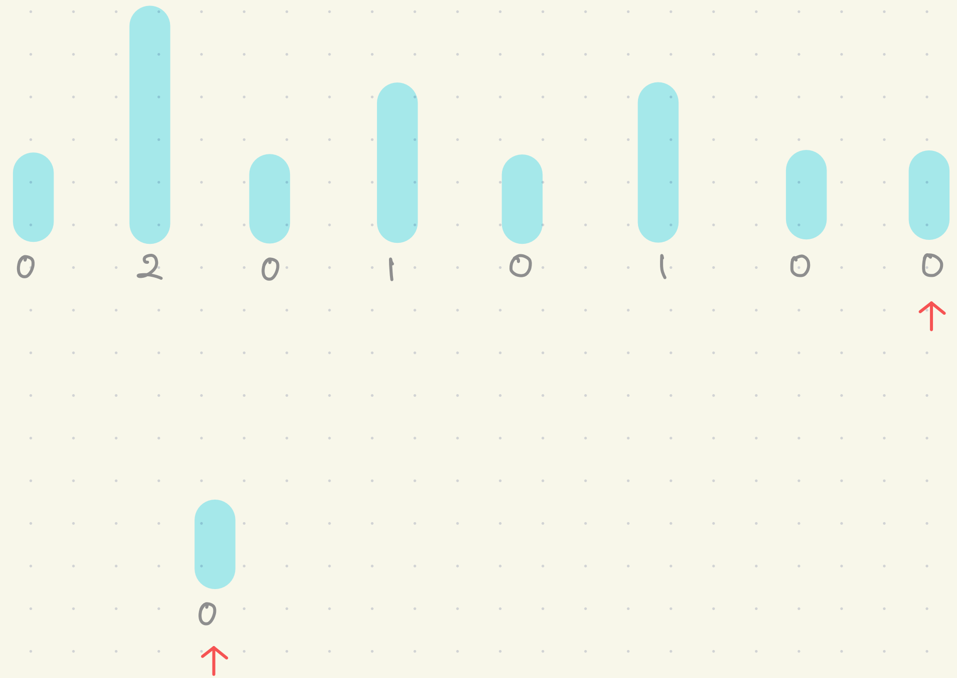
Transducer for Block Order



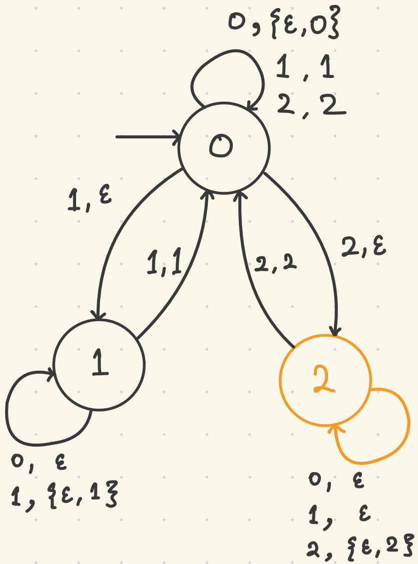


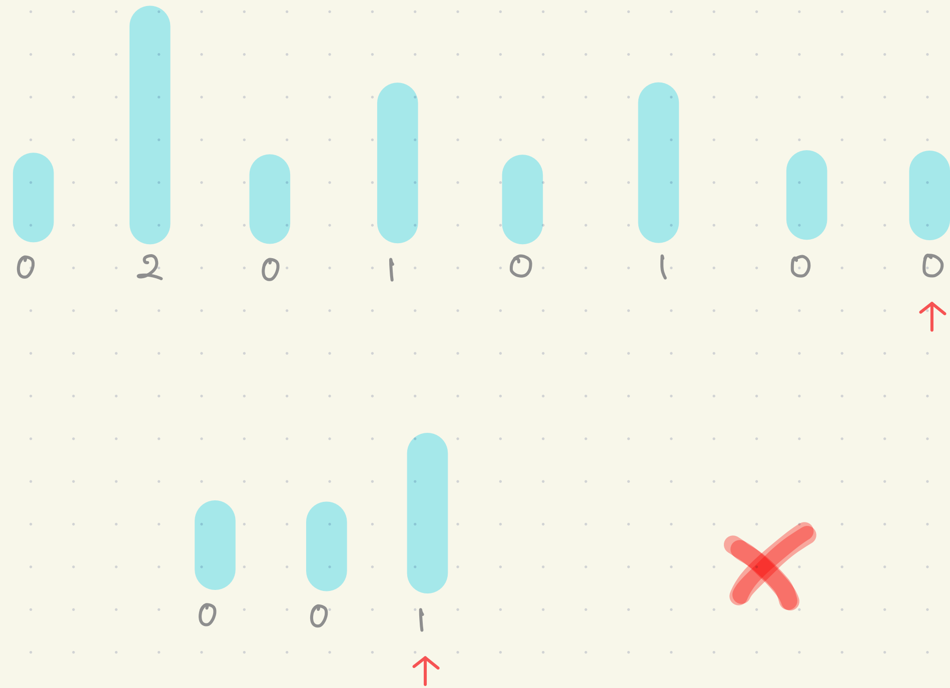
Transducer for Block Order



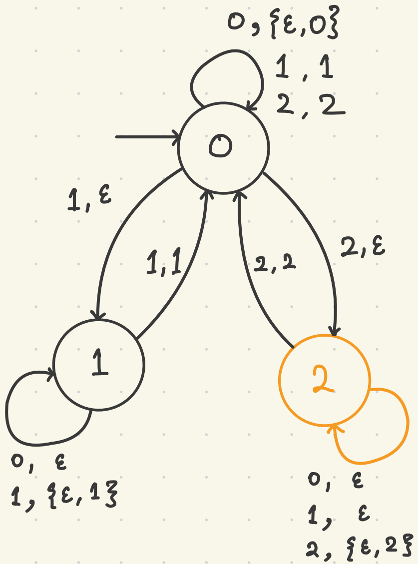


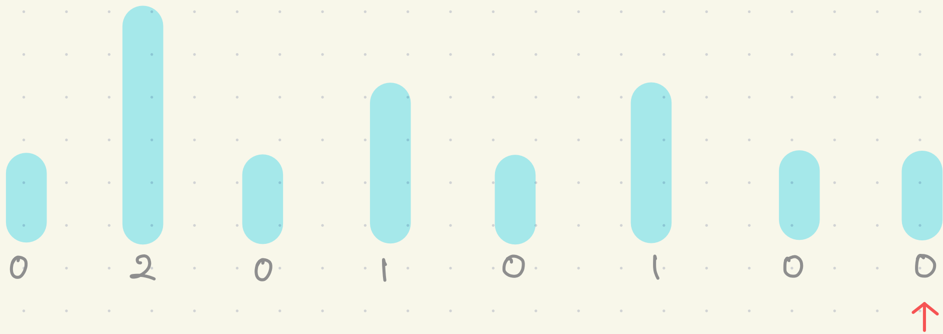
Transducer for Block Order





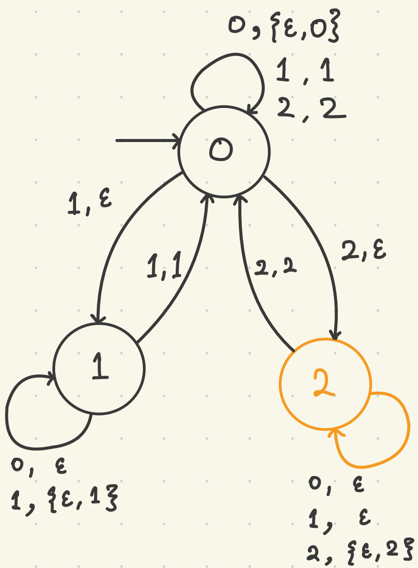
Transducer for Block Order

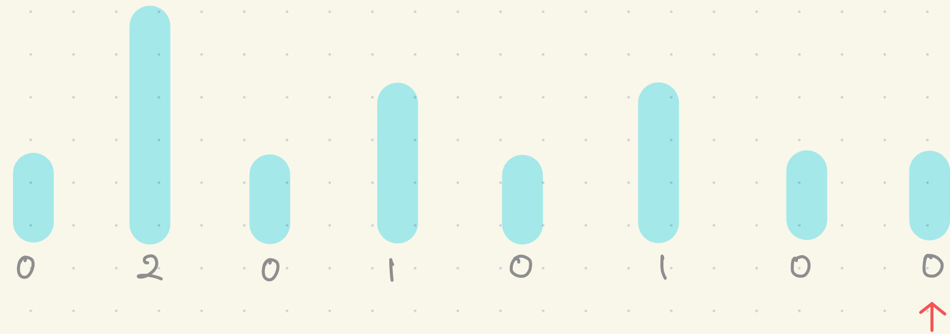




The transducer is computable in polytime.

Transducer for Block Order





The transducer is computable in polytime.

Downword closures for FSM can be computed in polytime.

One Counter Machines

FSA with
a counter
with zero
tests

One Counter Machines

FSA with
a counter
with zero
tests

FSA with
a counter
without
zero tests

One Counter Machines

FSA with
a counter
with zero
tests

FSA with
a counter
without
zero tests

FSA recognizing
block downward
closure.

One Counter Machines

FSA with a counter with zero tests

FSA with a counter *without* zero tests

FSA recognizing block downward closure.

- Keep track of the counter for a fixed polynomial bound.

One Counter Machines

FSA with a counter with zero tests

FSA with a counter *without* zero tests

FSA recognizing block downward closure.

- Keep track of the counter for a fixed polynomial bound.
- If bound is exceeded, there is a cycle which increases counter and one that decreases.

One Counter Machines

FSA with a counter with zero tests

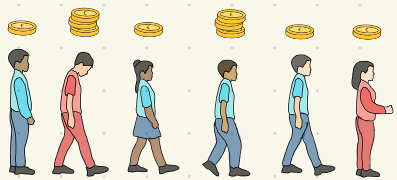
FSA with a counter without zero tests

FSA recognizing block downward closure.

Block downward closure for an OCA language can be computed in polytime.

Overview

Block Order



- Considers priorities
- Refines subword order and PSO

Simple Machines

- Downward closures computable in polytime

Pushdown Machines

Overview

Block Order

- Considers priorities
- Refines subword order and PSO

Simple Machines

- Downward closures computable in polytime

Pushdown Machines

Context Free Languages

$S \rightarrow ASB \mid \varepsilon$

$A \rightarrow 010$

$B \rightarrow 212$

Context Free Languages

S

$S \rightarrow ASB \mid \varepsilon$

$A \rightarrow 010$

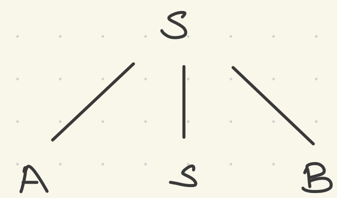
$B \rightarrow 212$

Context Free Languages

$S \rightarrow ASB \mid \varepsilon$

$A \rightarrow 010$

$B \rightarrow 212$

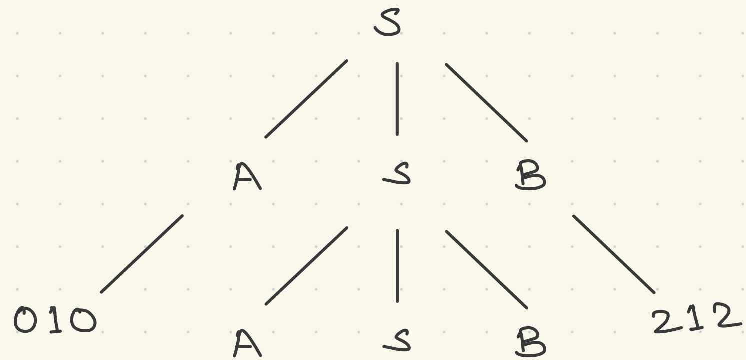


Context Free Languages

$S \rightarrow ASB \mid \varepsilon$

$A \rightarrow 010$

$B \rightarrow 212$

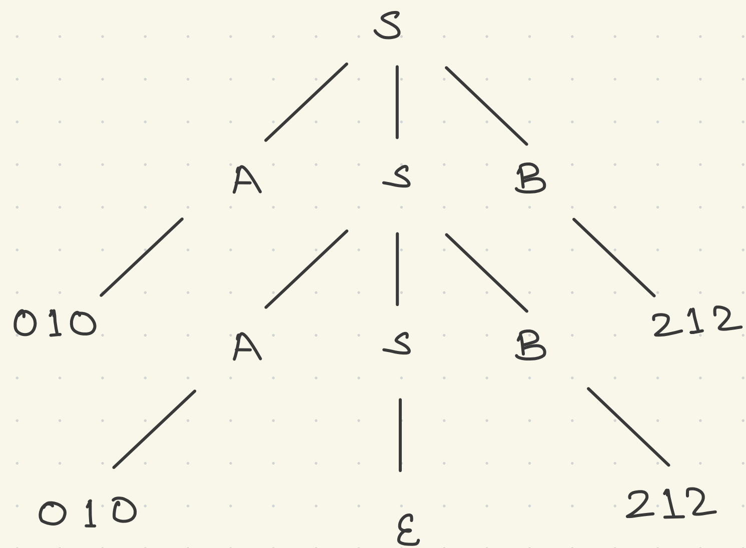


Context Free Languages

$S \rightarrow ASB \mid \varepsilon$

$A \rightarrow 010$

$B \rightarrow 212$



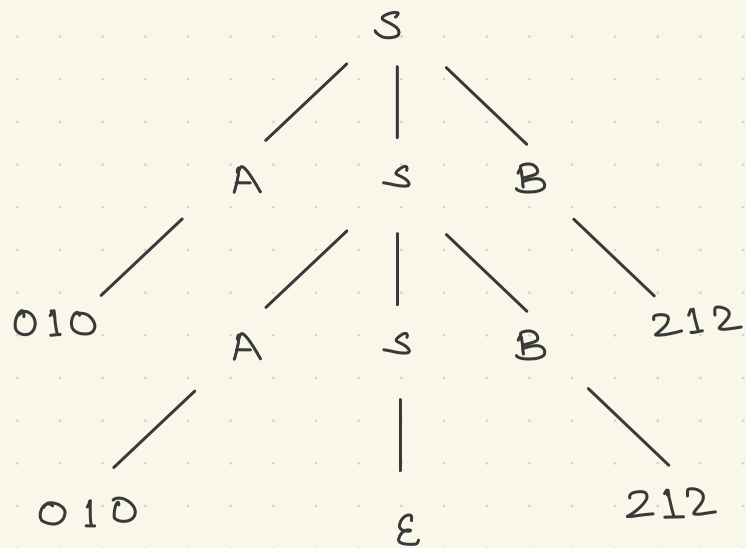
Context Free Languages

$S \longrightarrow ASB \mid \varepsilon$

$A \longrightarrow 010$

$B \longrightarrow 212$

$(010)^n (212)^n$



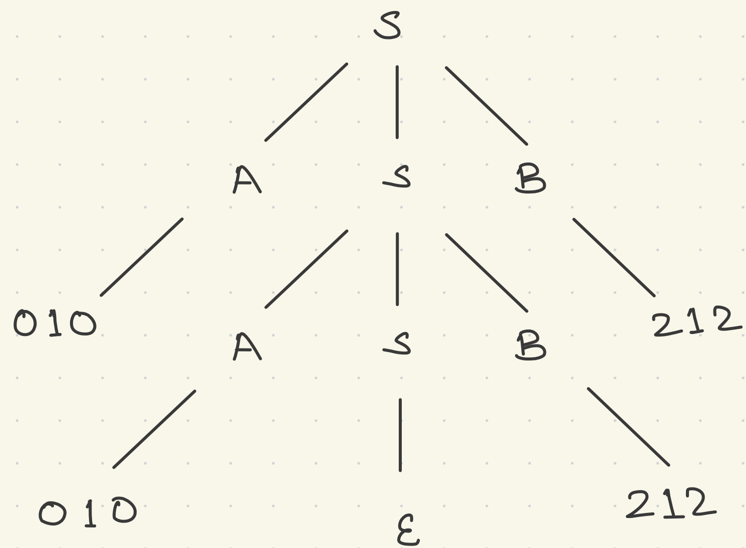
Context Free Languages

$S \rightarrow ASB \mid \varepsilon$

$A \rightarrow 010$

$B \rightarrow 212$

$(010)^n (212)^n$



Derivation tree can have arbitrary depth.

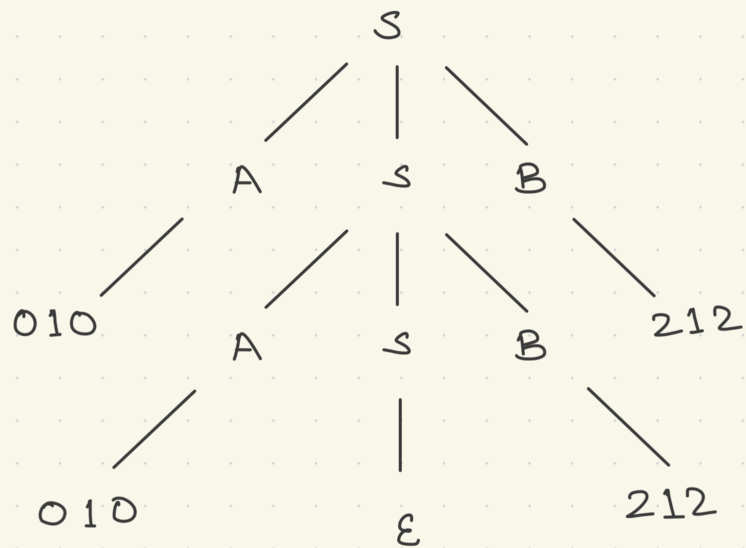
Context Free Languages

$S \to ASB \mid \varepsilon$

$A \to 010$

$B \to 212$

$(010)^n (212)^n$



Derivation tree can have arbitrary depth.

We try "bounding" the depth.

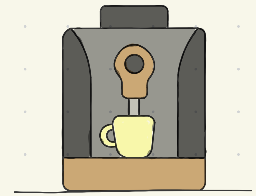
Context Free Languages

Context free

grammar

Context Free Languages

Context free grammar



Context Free Languages

Context free grammar

→

Another grammar that

- has same downward closure

Context Free Languages

Context free grammar

Another grammar that

- has same downward closure
- any word can be generated by "bounded" depth derivation trees.

Context Free Languages

$S \rightarrow ASB \mid \varepsilon$

$A \rightarrow 010$

$B \rightarrow 212$

$(010)^n (212)^n$

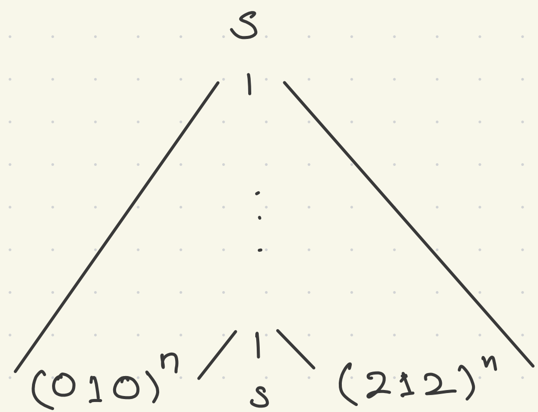
Context Free Languages

$S \rightarrow ASB \mid \varepsilon$

$A \rightarrow 010$

$B \rightarrow 212$

$(010)^n (212)^n$



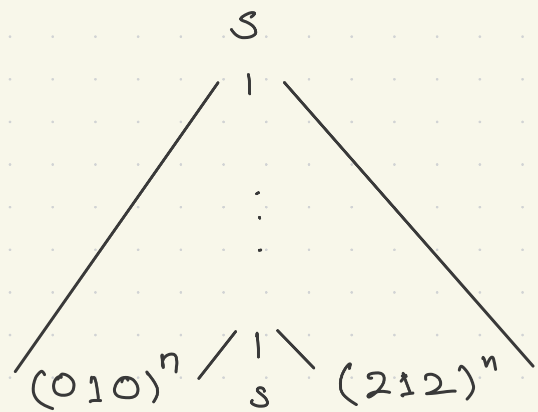
Context Free Languages

$S \rightarrow ASB \mid \varepsilon$

$A \rightarrow 010$

$B \rightarrow 212$

$(010)^n (212)^n$



Compute block downward closures of these cycles.

Context Free Languages

$$(010)^n \# (212)^n$$

Context Free Languages

$$(010)^n \# (212)^n$$

$(010)^n \Downarrow$

Context Free Languages

$$(010)^n \# (212)^n$$

$$(010)^n \Downarrow \;=\; L \Downarrow 1$$

set of first blocks

Context Free Languages

$$(010)^n \# (212)^n$$

$$(010)^n \Downarrow \;=\; \underbrace{L \Downarrow 1}_{\text{set of first blocks}} \cdot (\underbrace{M \Downarrow 1}_{\text{set of blocks surrounded by 1s}})^*$$

Context Free Languages

$$(010)^n \# (212)^n$$

$$(010)^n \Downarrow \;=\; L\Downarrow 1 \cdot (M\Downarrow 1)^* \cdot R\Downarrow$$

set of first blocks

set of blocks surrounded by 1s

set of last blocks

Context Free Languages

$$(010)^n \# (212)^n$$

$$(010)^n \Downarrow \;=\; \underbrace{L\Downarrow 1}_{\text{set of first blocks}} \cdot \underbrace{(M\Downarrow 1)^*}_{\text{set of blocks surrounded by 1s}} \cdot \underbrace{R\Downarrow}_{\text{set of last blocks}}$$

L, M, R have one less priority

Context Free Languages

$$(010)^n \# (212)^n$$

$$(010)^n \Downarrow \; = \; L\Downarrow 1 \cdot (M\Downarrow 1)^* \cdot R\Downarrow$$

set of first blocks

set of blocks surrounded by 1s

set of last blocks

$$= \{\varepsilon, 0\} 1 \cdot (\{\varepsilon, 0, 00\} 1)^* \cdot \{\varepsilon, 0\}$$

Context Free Languages

$$(010)^n \# (212)^n$$

$$(010)^n \Downarrow \;=\; \underbrace{L\Downarrow}_{\text{set of first blocks}} 2 \cdot (\underbrace{M\Downarrow}_{\text{set of blocks surrounded by 2s}} 2)^* \cdot \underbrace{R\Downarrow}_{\text{set of last blocks}}$$

$$= \{\varepsilon\}\, 2 \cdot (\{\varepsilon, 1\}\, 2)^* \cdot \{\varepsilon\}$$

Context Free Languages

$S \to ASB \mid \varepsilon$

$A \to 010$

$B \to 212$

$\longrightarrow$

Another grammar that

- has same downward closure
- any word can be generated by "bounded" depth derivation trees.

Context Free Languages

$$S \rightarrow ASB \mid \varepsilon$$
$$A \rightarrow 010$$
$$B \rightarrow 212$$

$\longrightarrow$

$$S \rightarrow ASB \mid \varepsilon$$
$$A \rightarrow 010$$
$$B \rightarrow 212$$

Context Free Languages

$$S \to ASB \mid \varepsilon$$
$$A \to 010$$
$$B \to 212$$

$$\longrightarrow$$

$$S \to ASB \mid \varepsilon \mid \bar{A}^* \# \bar{B}^*$$
$$A \to 010$$
$$B \to 212$$
$$\bar{A} \to \{\varepsilon, 0\}\, 1\, (\{\varepsilon, 0, 00\} \cdot 1)^*\, \{\varepsilon, 0\}$$
$$\bar{B} \to 2 \cdot (\{\varepsilon, 1\} \cdot 2)^*$$

Context Free Languages

$$S \to ASB \mid \varepsilon$$

$$A \to 010$$

$$B \to 212$$

$$\longrightarrow$$

$$S \to ASB \mid \varepsilon \mid \bar{A}^* \# \bar{B}^*$$

$$A \to 010$$

$$B \to 212$$

$$\bar{A} \to \{\varepsilon, 0\}\, 1\, (\{\varepsilon, 0, 00\} \cdot 1)^* \{\varepsilon, 0\}$$

$$\bar{B} \to 2 \cdot (\{\varepsilon, 1\} \cdot 2)^*$$

$$\# \to \varepsilon$$

Context Free Languages

$$S \to ASB \mid \varepsilon$$
$$A \to 010$$
$$B \to 212$$

$\longrightarrow$

$$S \to ASB \mid \varepsilon \mid \bar{A}^* \# \bar{B}^*$$
$$A \to 010$$
$$B \to 212$$
$$\bar{A} \to \{\varepsilon, 0\} 1 (\{\varepsilon, 0, 00\} \cdot 1)^* \{\varepsilon, 0\}$$
$$\bar{B} \to 2 \cdot (\{\varepsilon, 1\} \cdot 2)^*$$
$$\# \to \varepsilon$$

Both grammars have same block downward closures.

Context Free Languages

$S \to ASB \mid \varepsilon$

$A \to 010$

$B \to 212$

$\longrightarrow$

Exponential blowup.

$S \to ASB \mid \varepsilon \mid \bar{A}^* \# \bar{B}^*$

$A \to 010$

$B \to 212$

$\bar{A} \to \{\varepsilon, 0\}\, 1\, (\{\varepsilon, 0, 00\} \cdot 1)^* \{\varepsilon, 0\}$

$\bar{B} \to 2 \cdot (\{\varepsilon, 1\} \cdot 2)^*$

$\# \to \varepsilon$

Context Free Languages

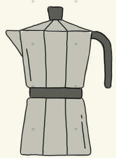
$$
\begin{aligned}
S &\to ASB \mid \varepsilon \\
A &\to 010 \\
B &\to 212
\end{aligned}
$$

Exponential blowup.

$$
\begin{aligned}
S &\to ASB \mid \varepsilon \mid \bar{A}^* \# \bar{B}^* \\
A &\to 010 \\
B &\to 212 \\
\bar{A} &\to \{\varepsilon, 0\}\, 1\, (\{\varepsilon, 0, 00\} \cdot 1)^* \{\varepsilon, 0\} \\
\bar{B} &\to 2 \cdot (\{\varepsilon, 1\} \cdot 2)^* \\
\# &\to \varepsilon
\end{aligned}
$$

Can be recognized by a finite state machine of exponential size.

Context Free Languages



Downward closure of a CFG can be recognized by a FSM of doubly exponential size.

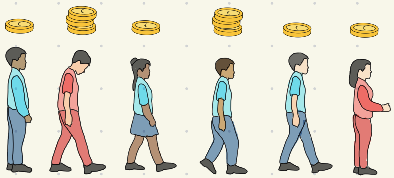
Context Free Languages

- Downward closure of a CFG can be recognized by a FSM of doubly exponential size.

- Exponential lower bound is inherited from subword order.

Summary

Block Order



- Considers priorities
- Refines subword order and PSO

Simple Machines

- Downward closures computable in polytime

Pushdown Machines

- 2-EXP upper bound
- EXP lower bound