

# Lecture 13

## 4.1.4 Simplified Turing Machine Models

## 4.1.5 Generating Languages Using Turing Machines

# Lecture 13

## 4.1.4 Simplified Turing Machine Models

## 4.1.5 Generating Languages Using Turing Machines

# Simplified TM Models

- **Stack machine**
- **Counter machines**
- **TM with a limited number of states and tape symbols**
- **Universal Turing Machine**

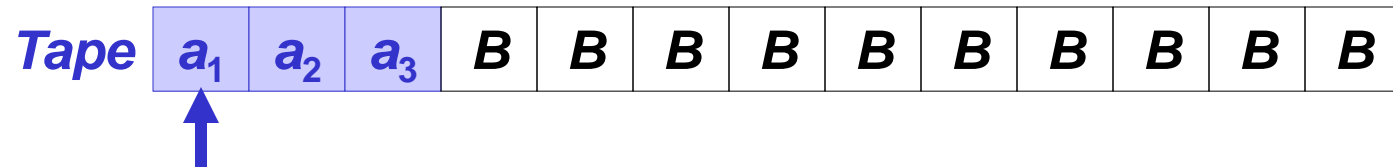
# Stack Machine

# Stack Machine

*Turing machine*

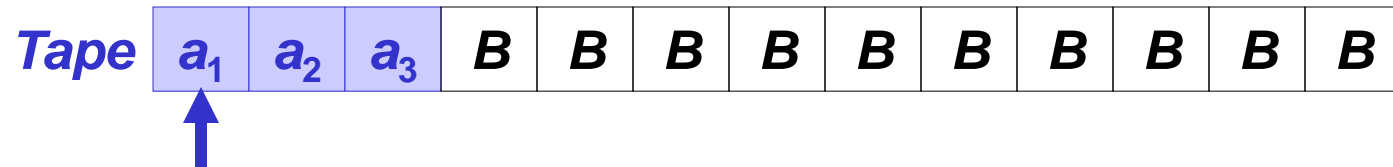
# Stack Machine

*Turing machine*



# Stack Machine

*Turing machine*

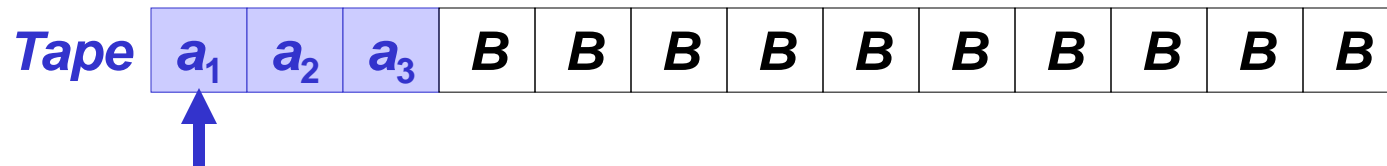


---

*Stack machine*

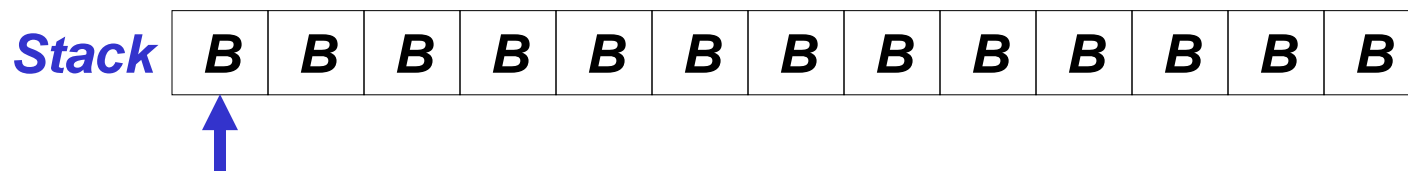
# Stack Machine

## *Turing machine*



---

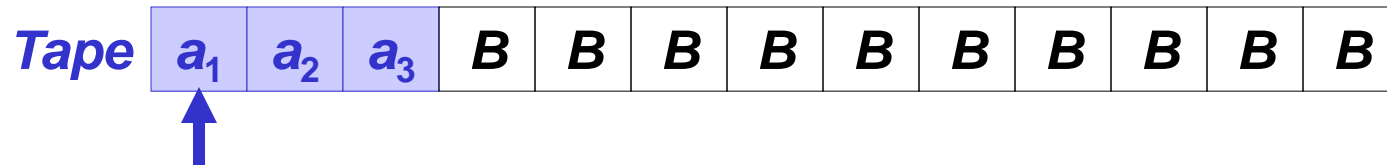
## *Stack machine*



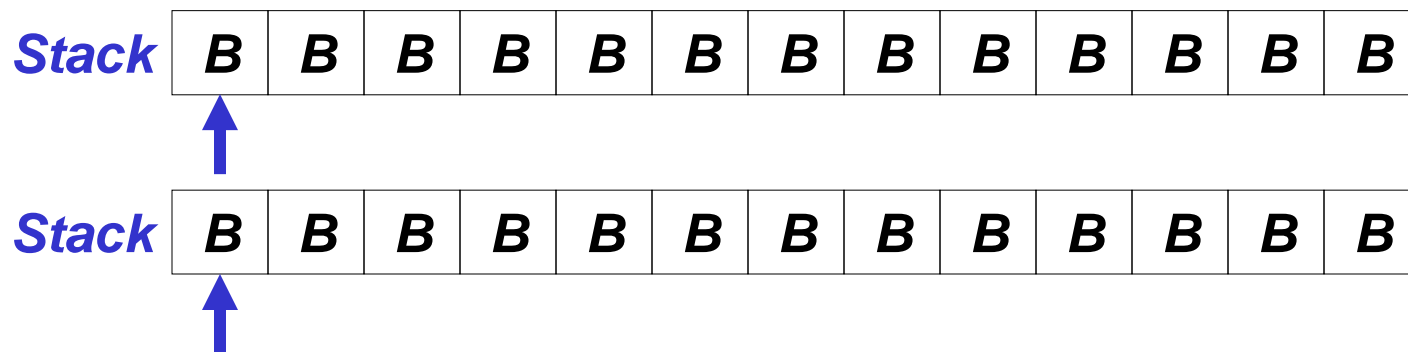


# Stack Machine

## *Turing machine*

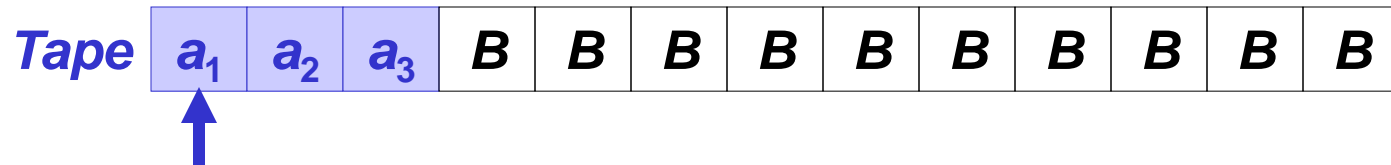


## *Stack machine*

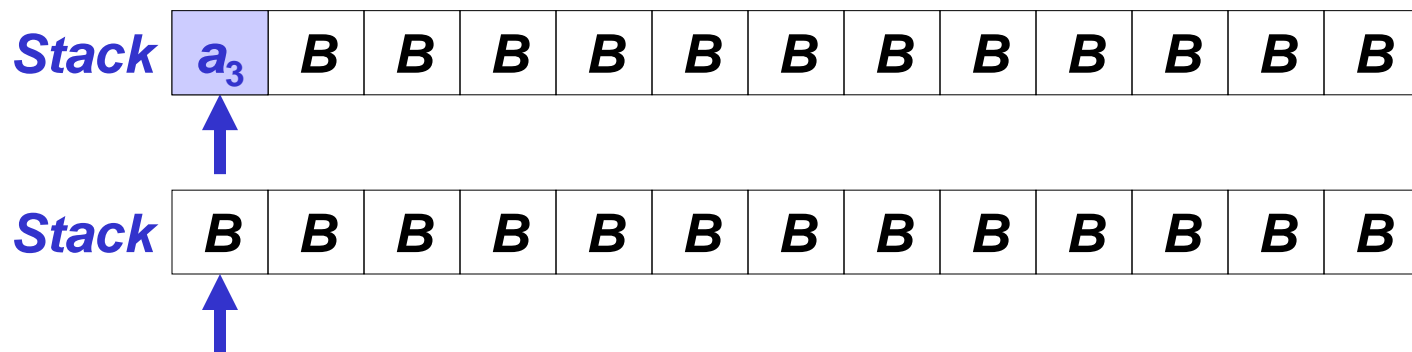


# Stack Machine

## *Turing machine*

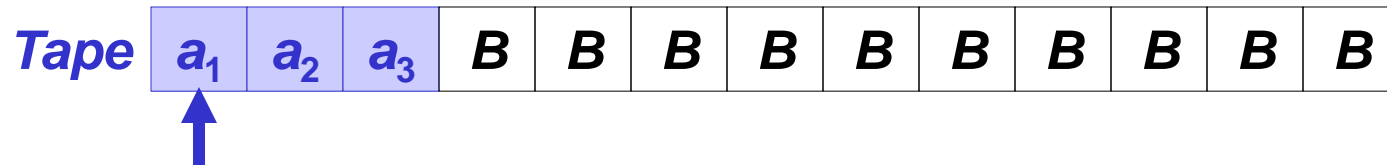


## *Stack machine*

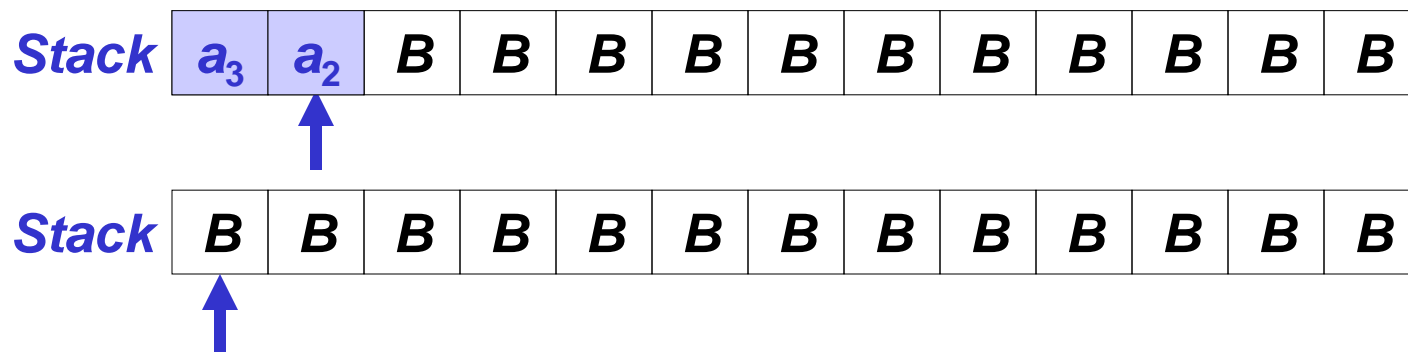


# Stack Machine

## *Turing machine*

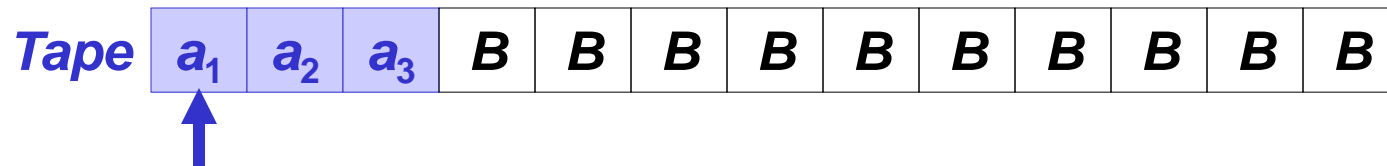


## *Stack machine*

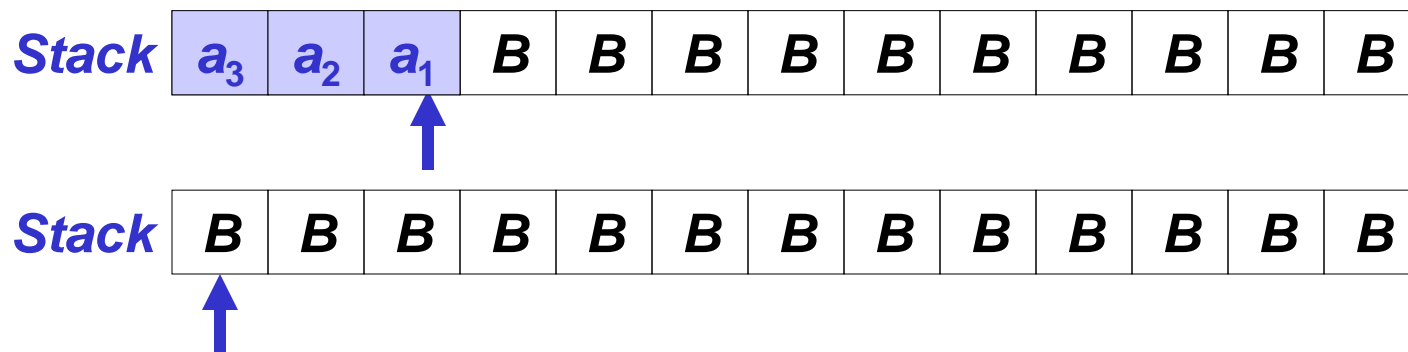


# Stack Machine

## Turing machine

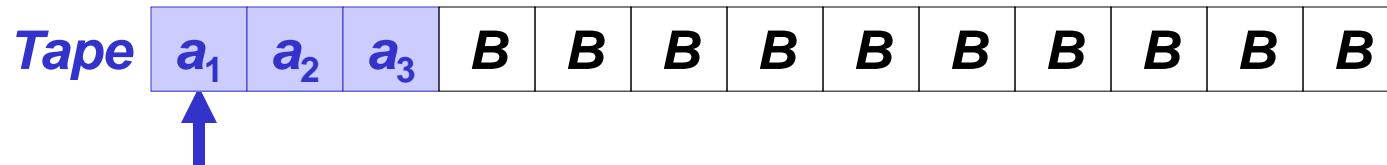


## Stack machine

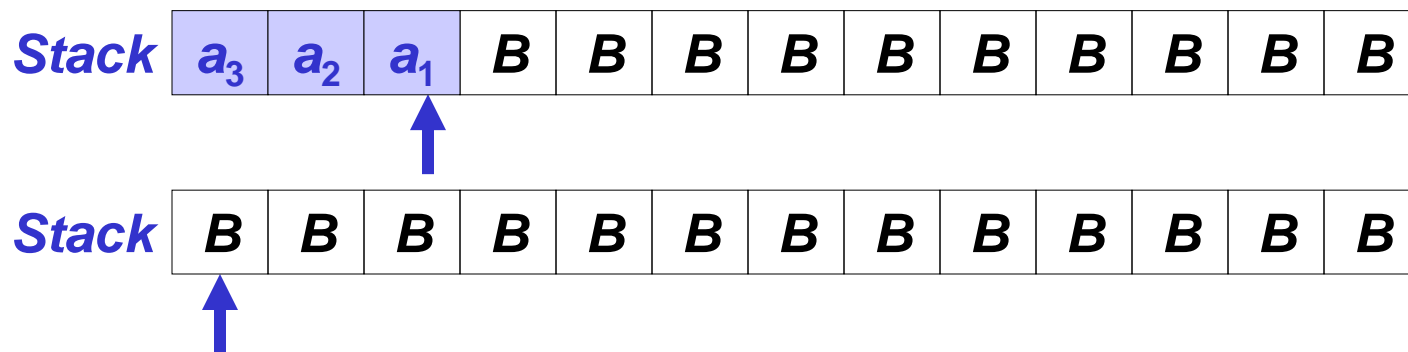


# Stack Machine

## Turing machine

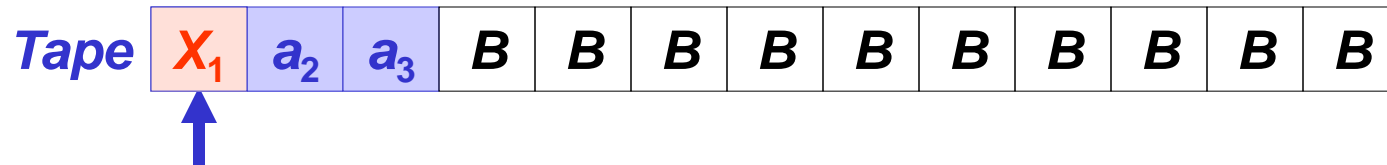


## Stack machine

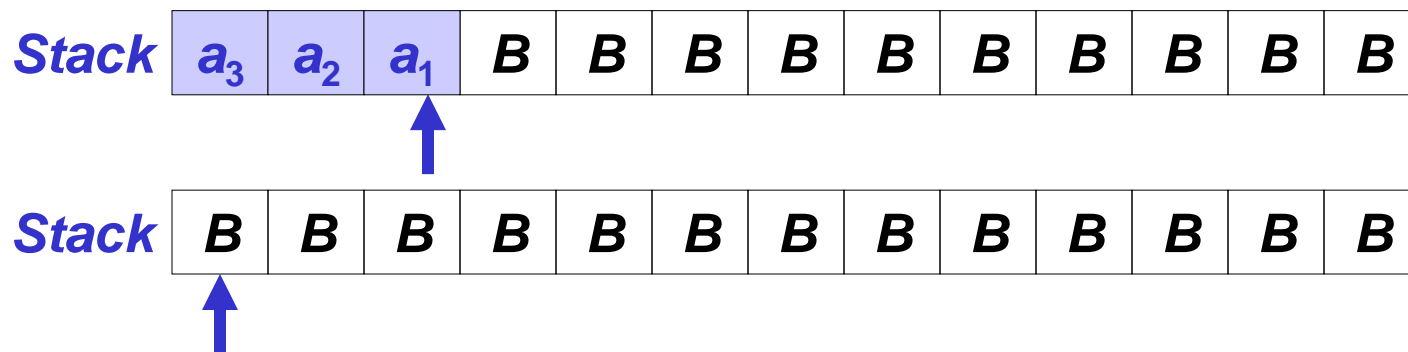


# Stack Machine

## *Turing machine*

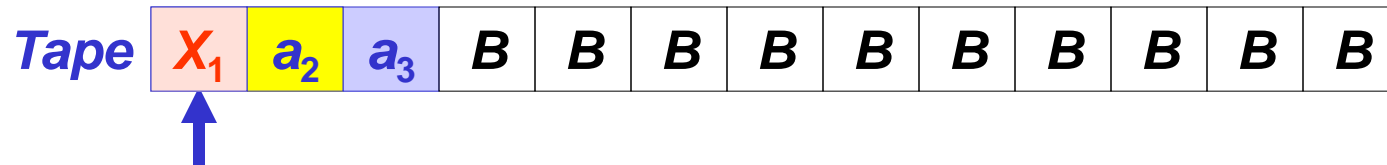


## *Stack machine*

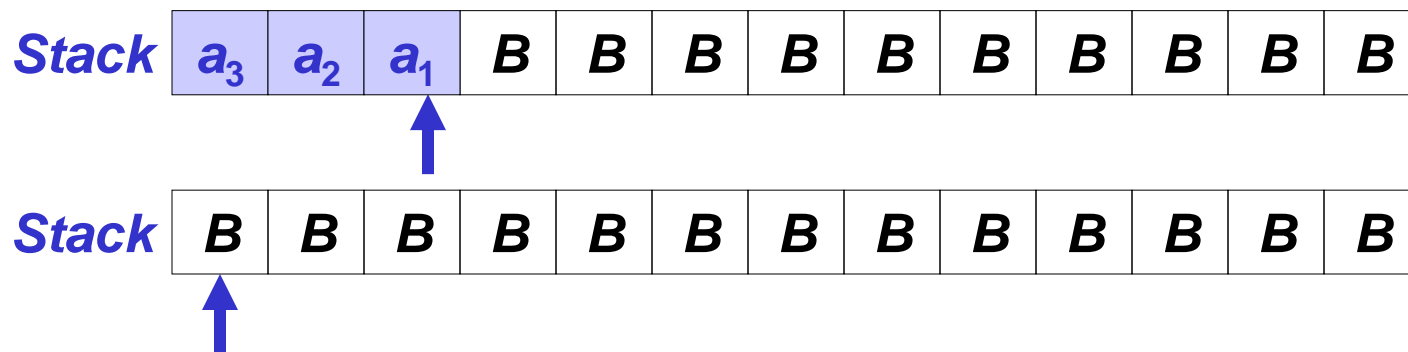


# Stack Machine

## *Turing machine*

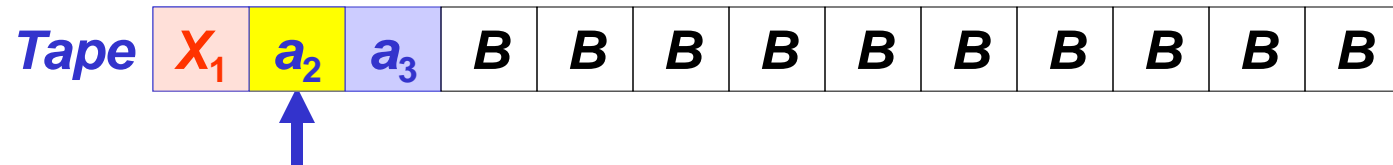


## *Stack machine*

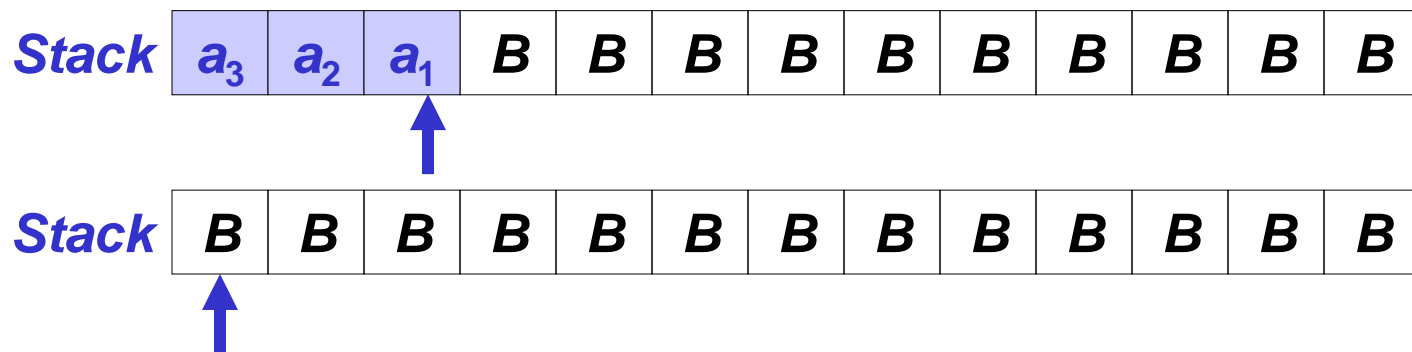


# Stack Machine

## Turing machine



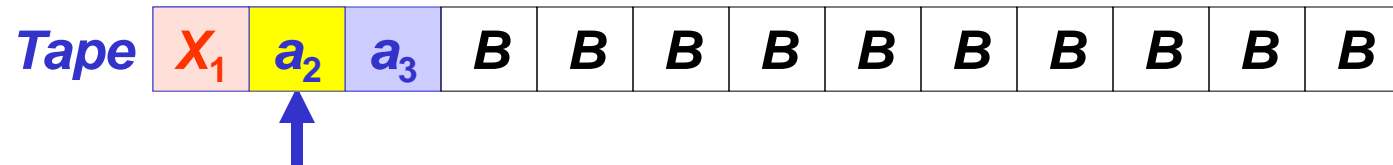
## Stack machine



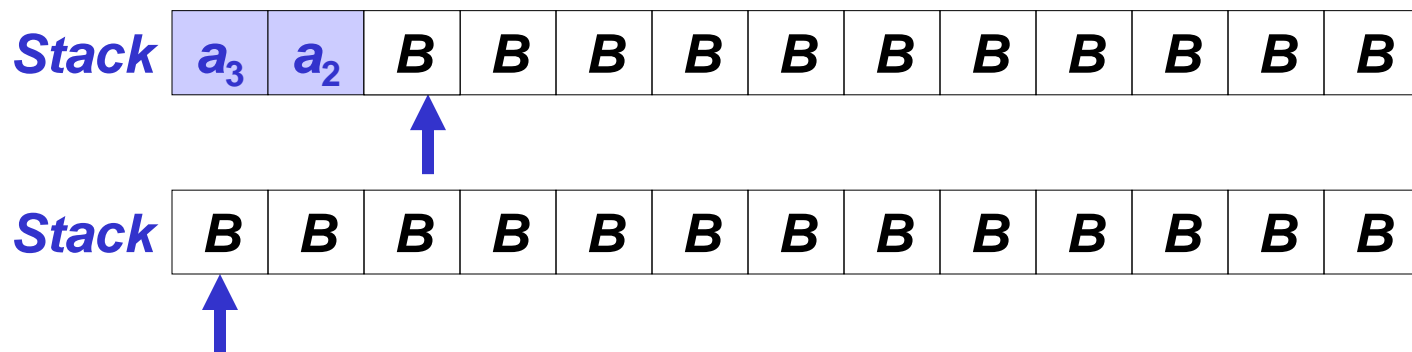


# Stack Machine

## *Turing machine*

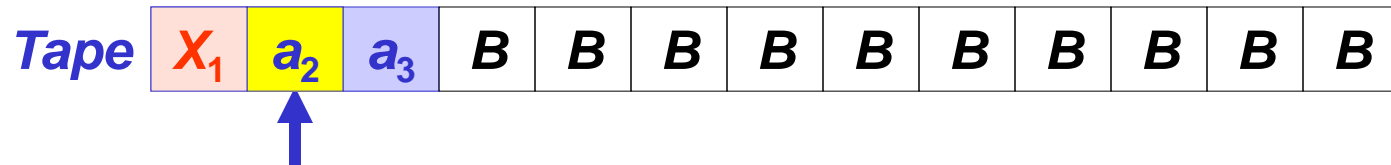


## *Stack machine*

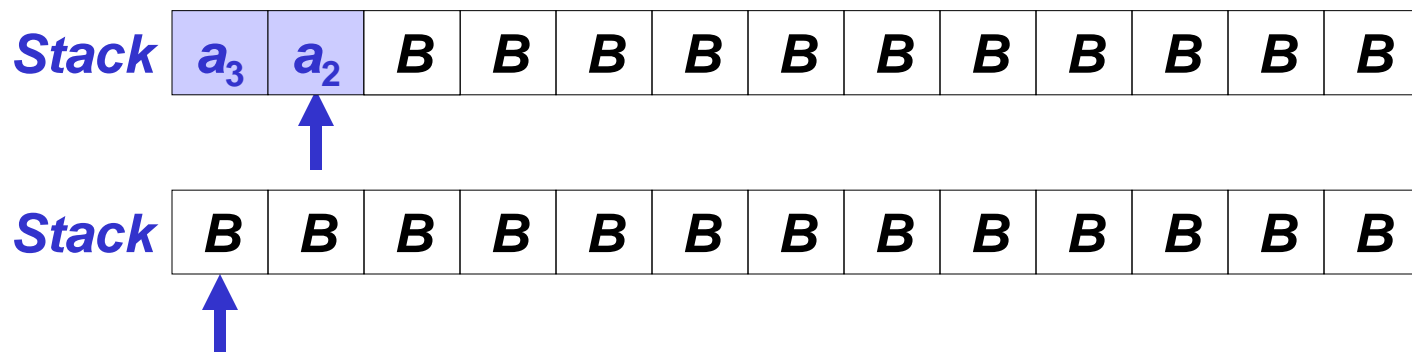


# Stack Machine

## Turing machine

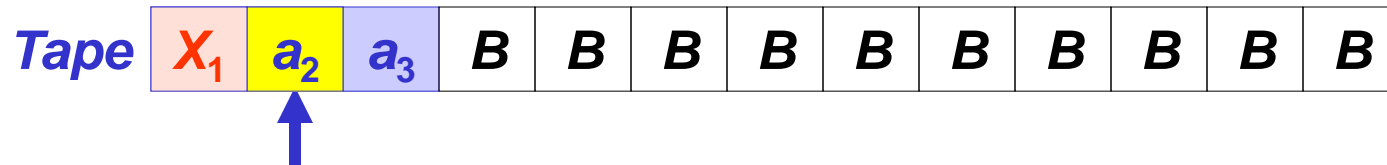


## Stack machine

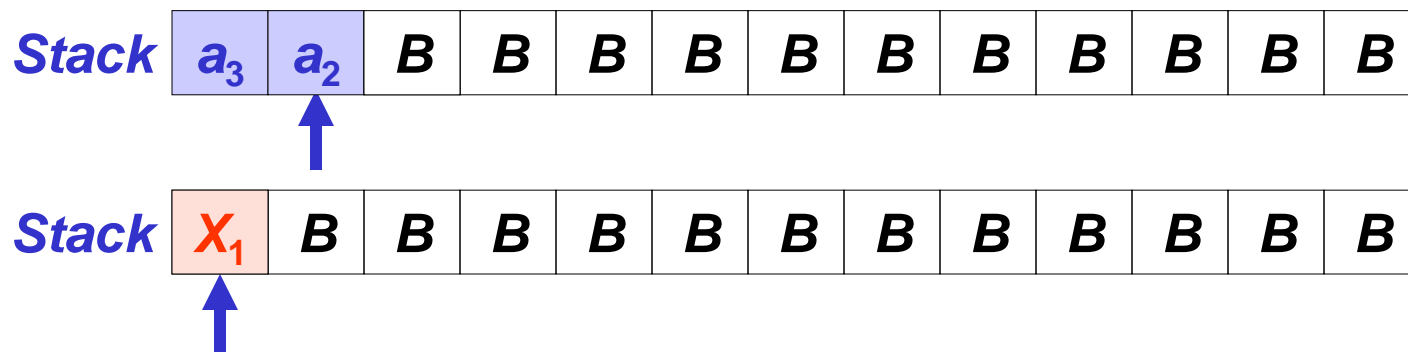


# Stack Machine

## *Turing machine*

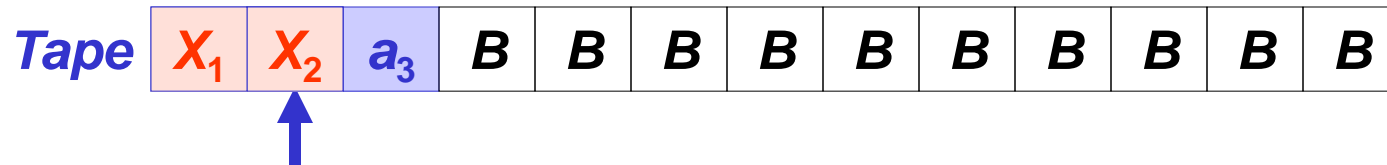


## *Stack machine*

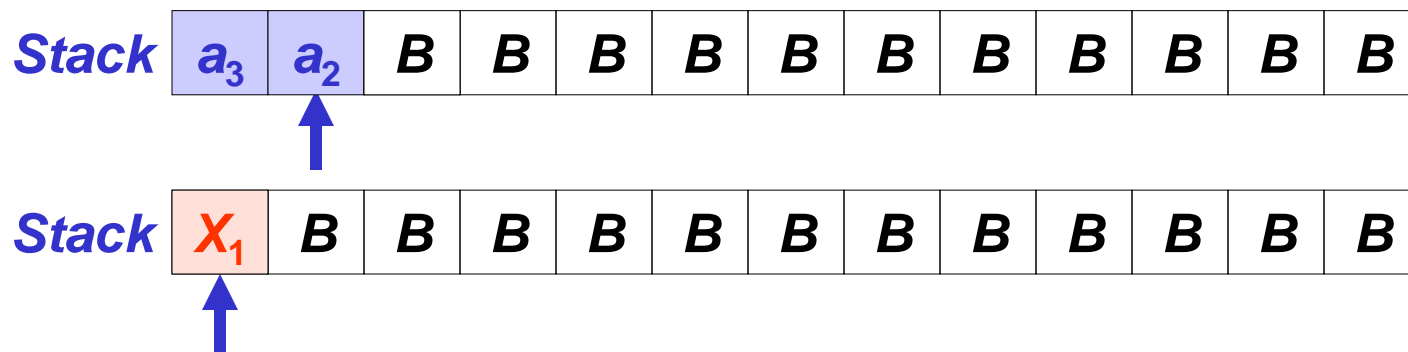


# Stack Machine

## *Turing machine*

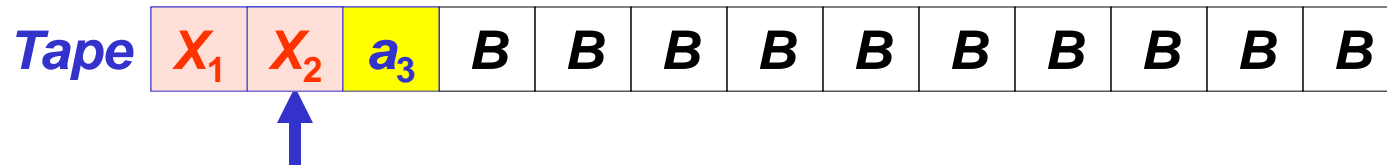


## *Stack machine*

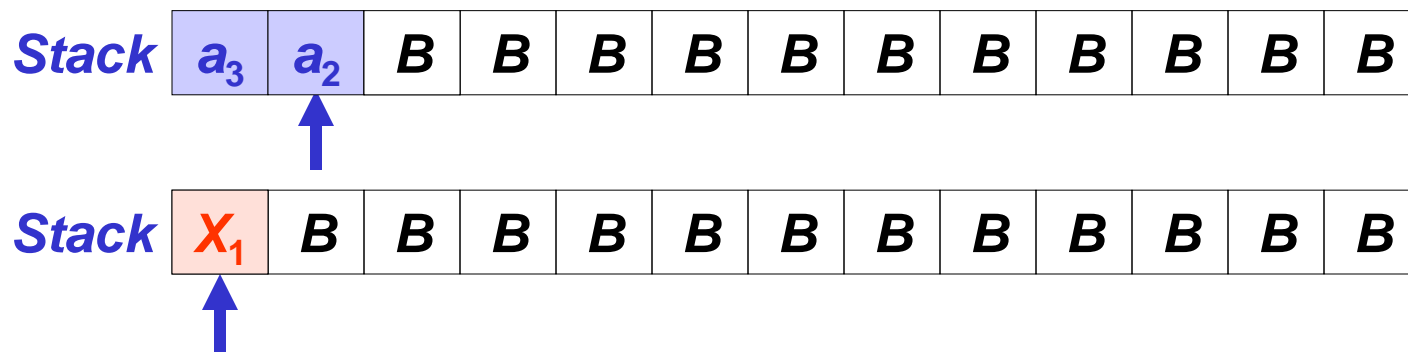


# Stack Machine

## *Turing machine*

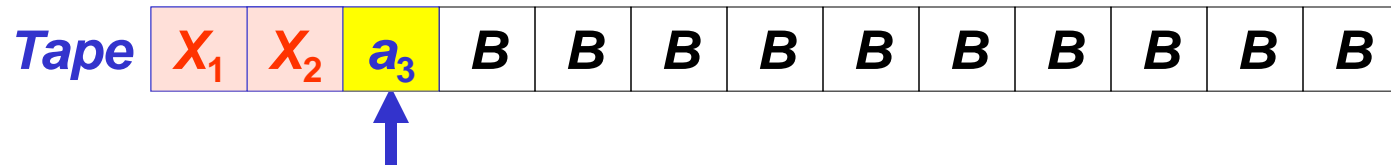


## *Stack machine*

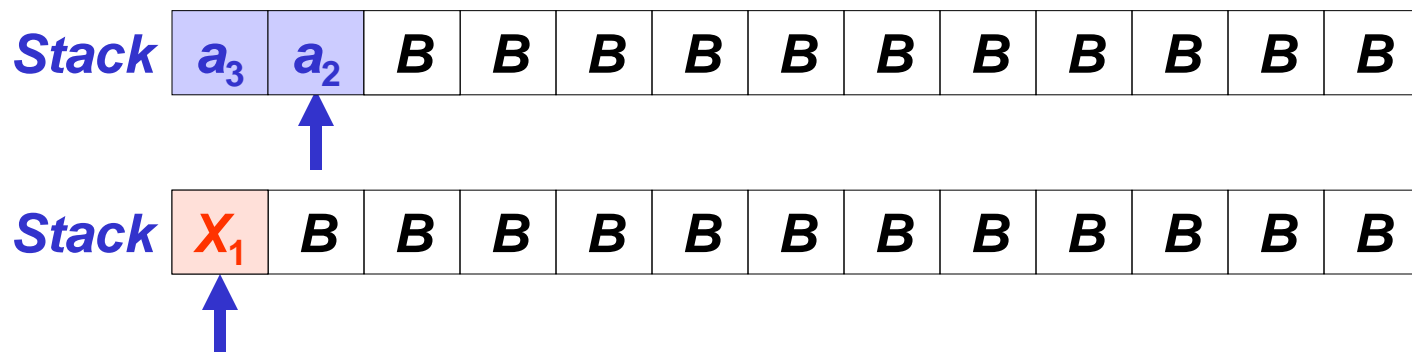


# Stack Machine

## *Turing machine*

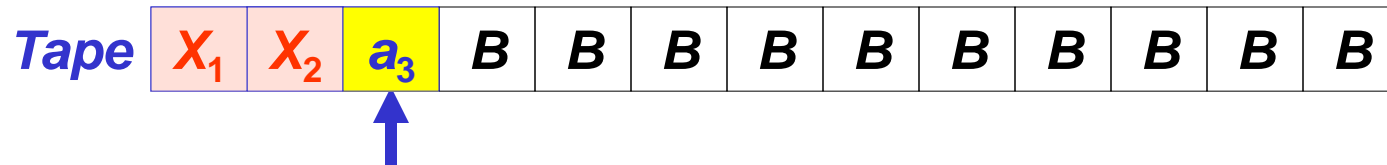


## *Stack machine*

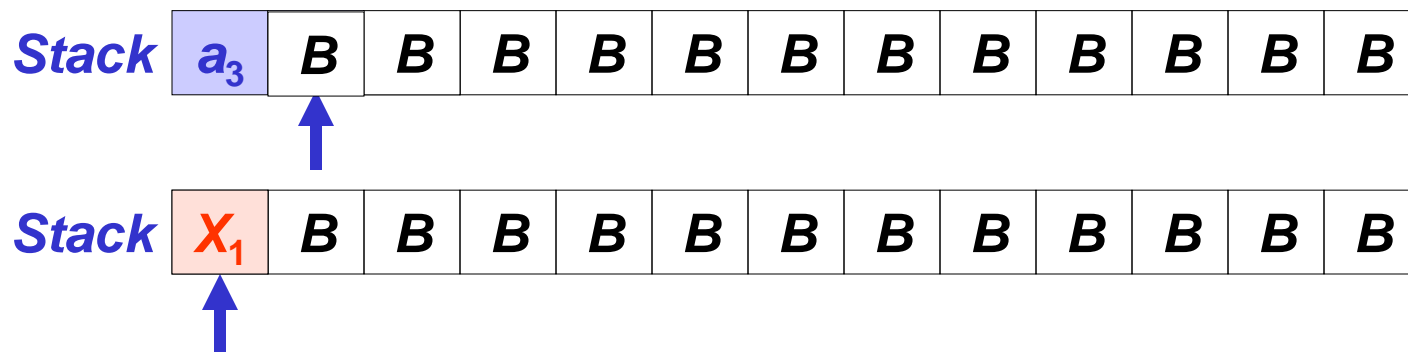


# Stack Machine

## *Turing machine*

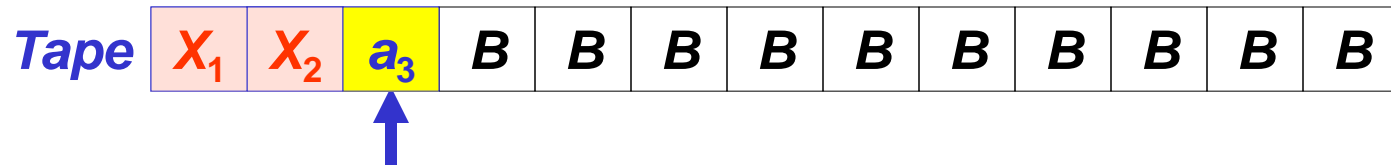


## *Stack machine*

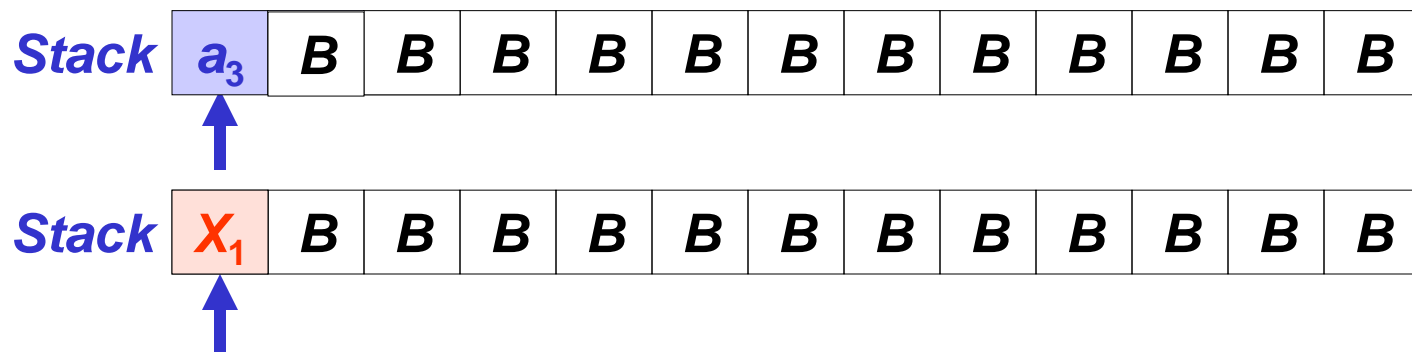


# Stack Machine

## Turing machine



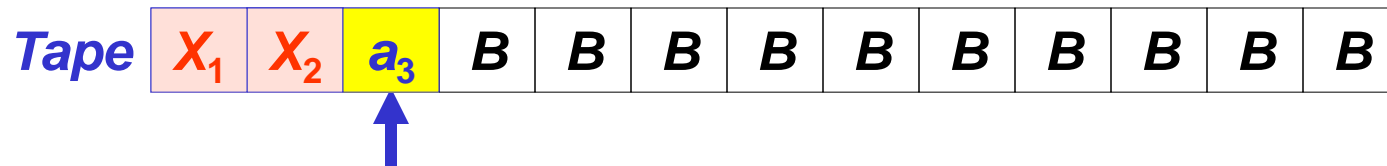
## Stack machine



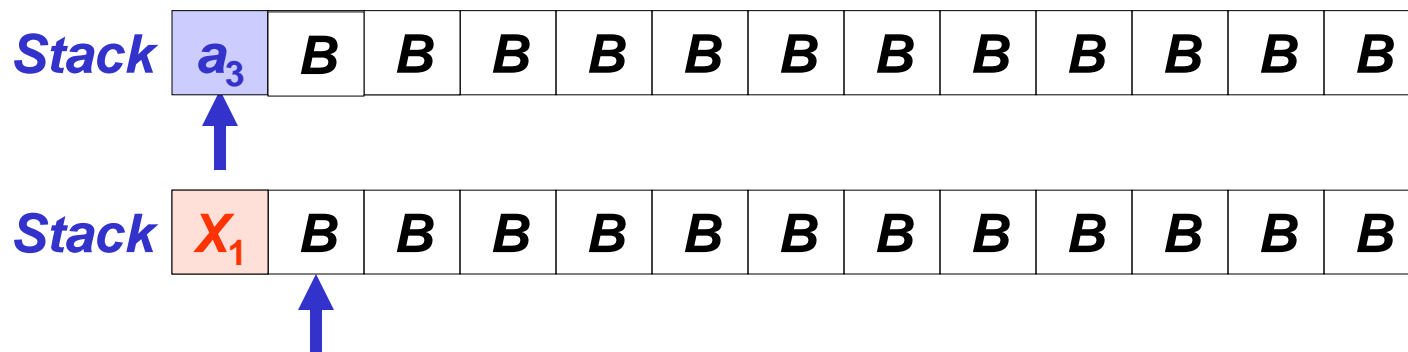


# Stack Machine

## *Turing machine*

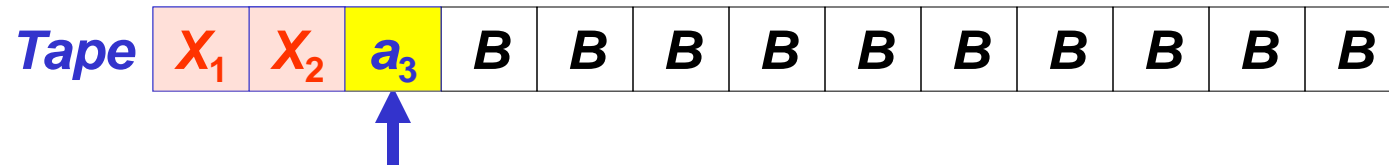


## *Stack machine*

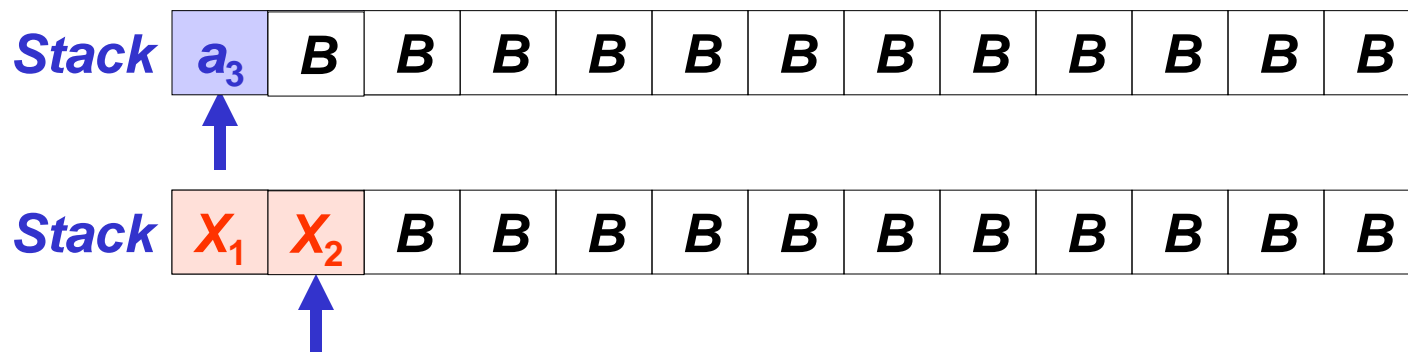


# Stack Machine

## Turing machine

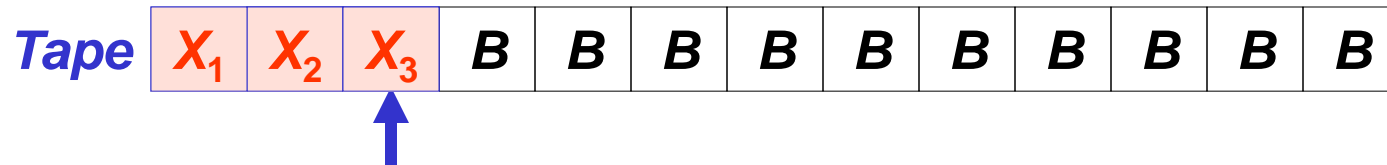


## Stack machine

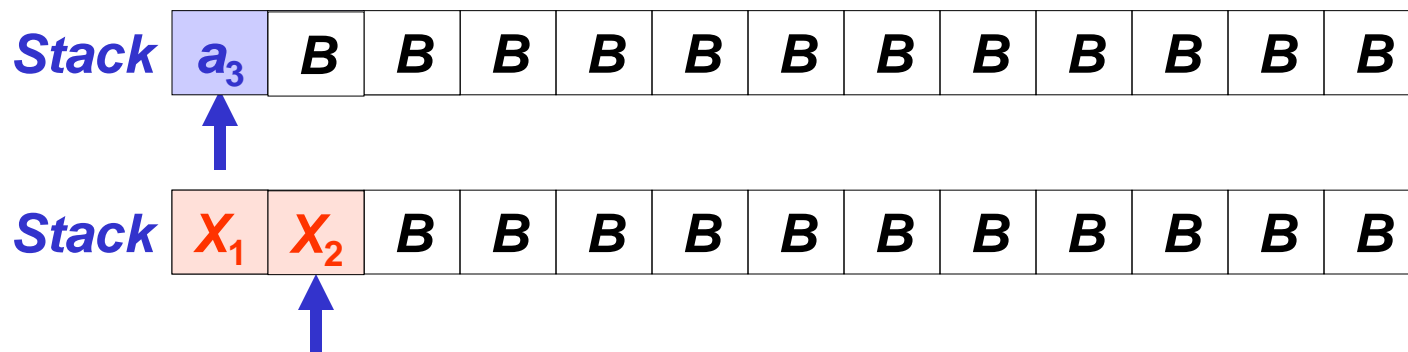


# Stack Machine

## *Turing machine*

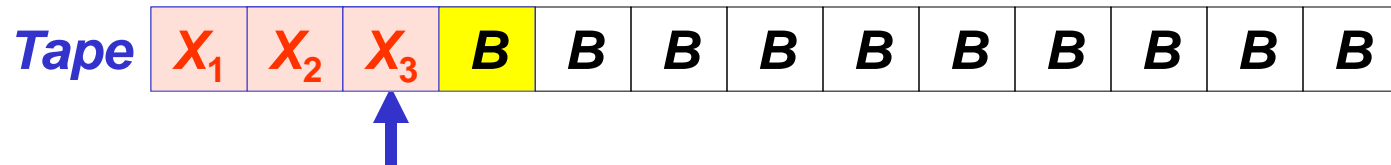


## *Stack machine*

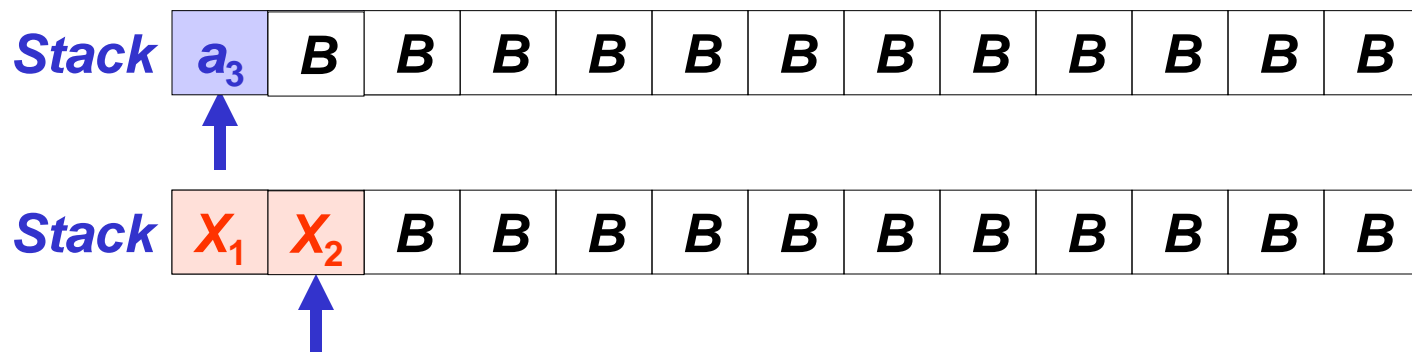


# Stack Machine

## Turing machine

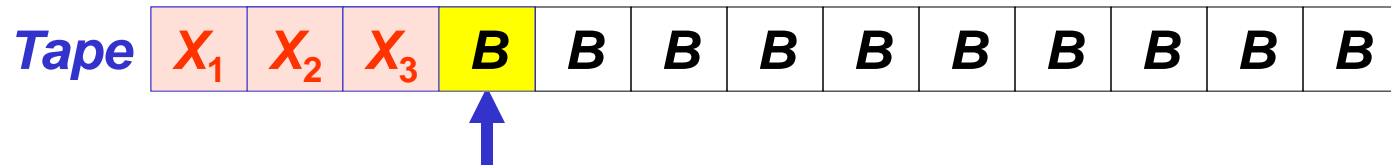


## Stack machine

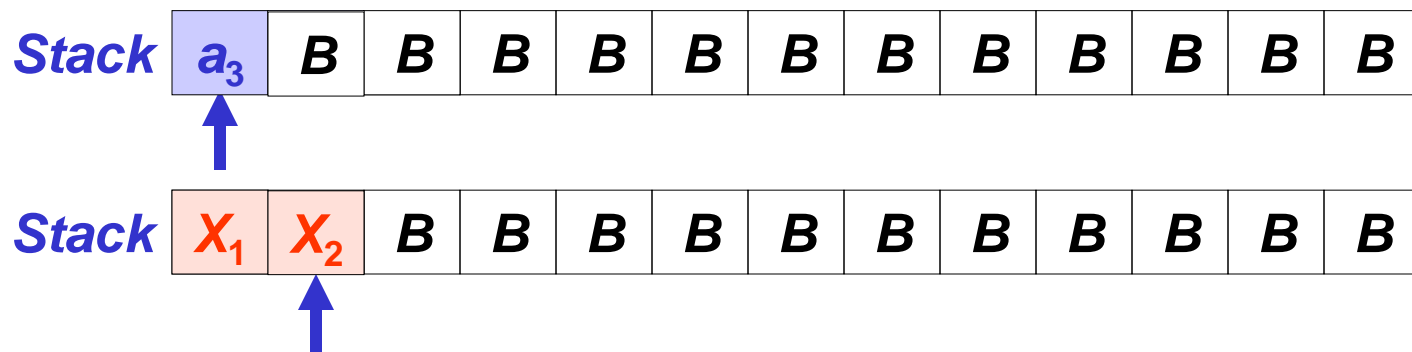


# Stack Machine

## Turing machine

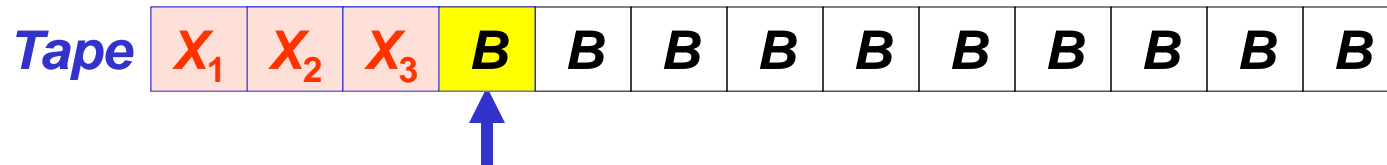


## Stack machine

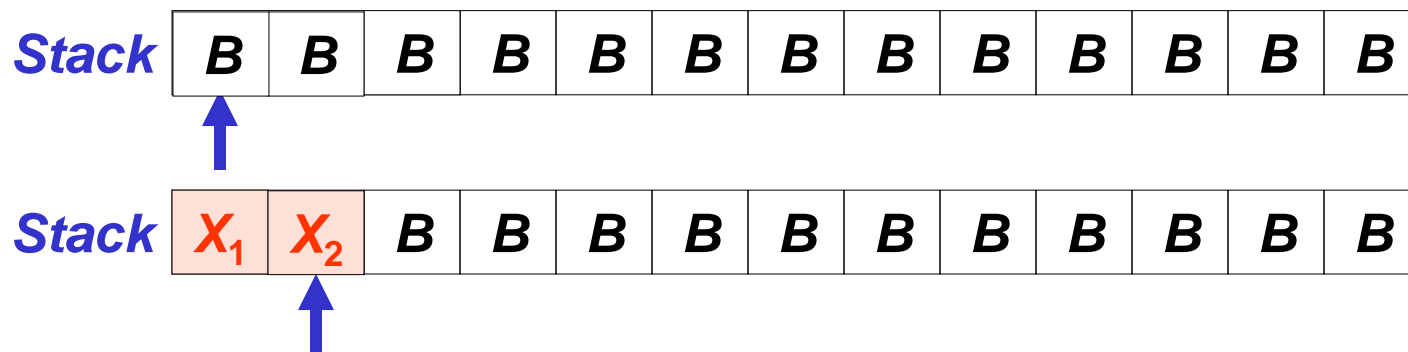


# Stack Machine

## *Turing machine*

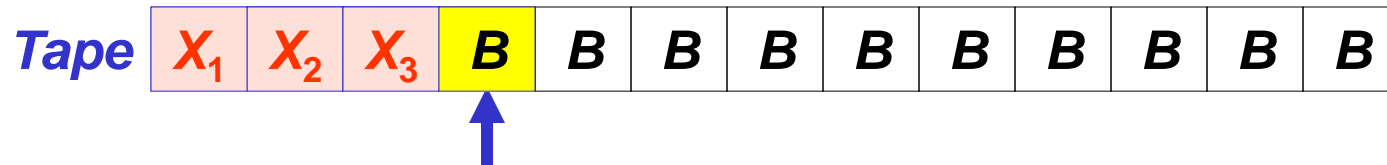


## *Stack machine*

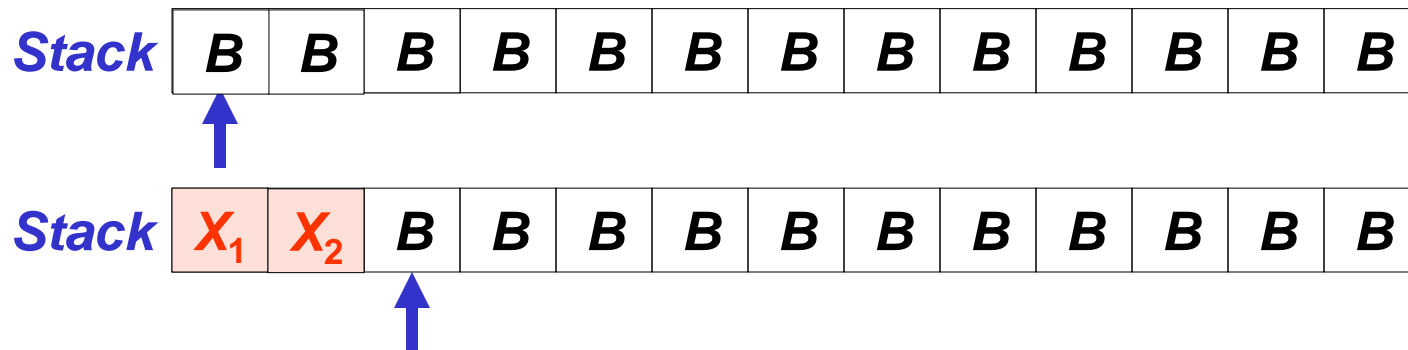


# Stack Machine

## Turing machine

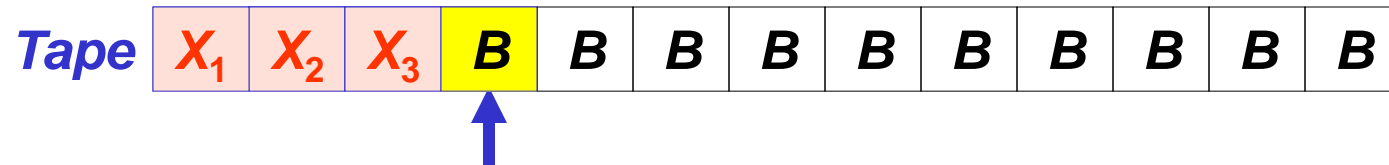


## Stack machine

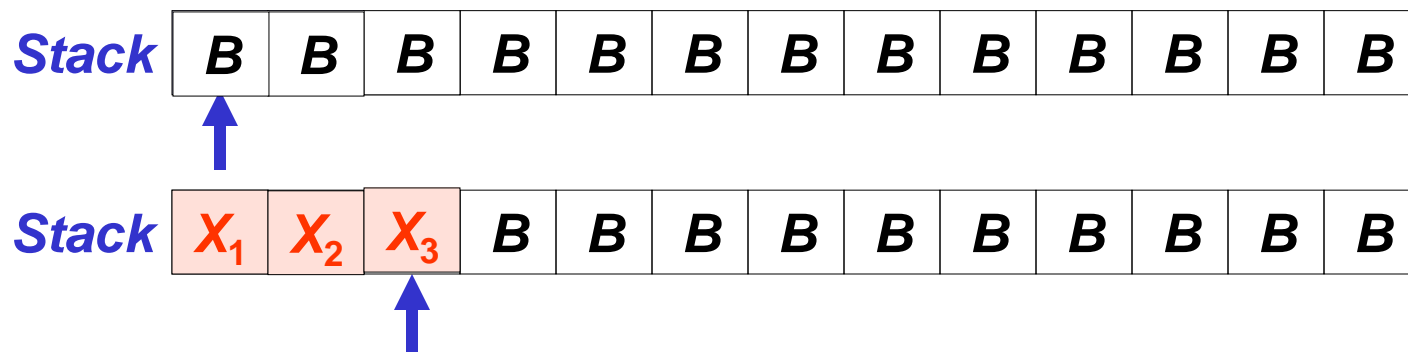


# Stack Machine

## *Turing machine*



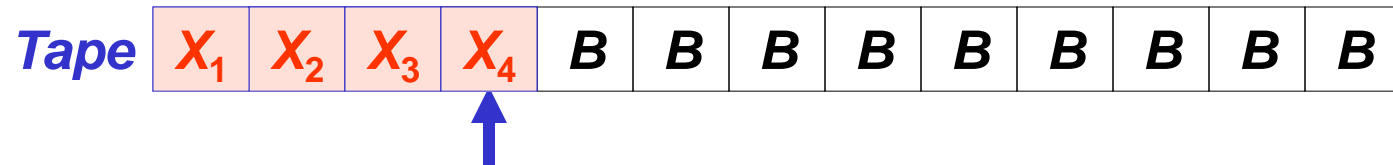
## *Stack machine*



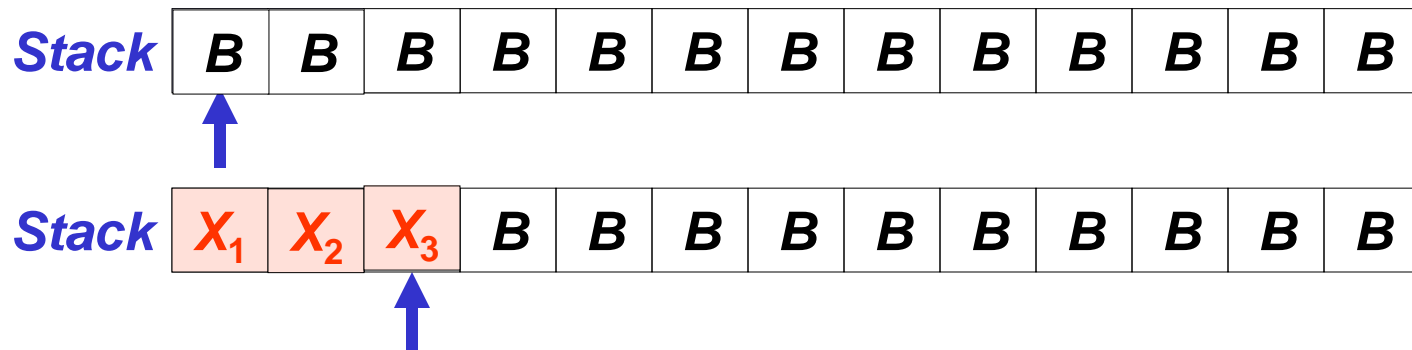


# Stack Machine

## *Turing machine*

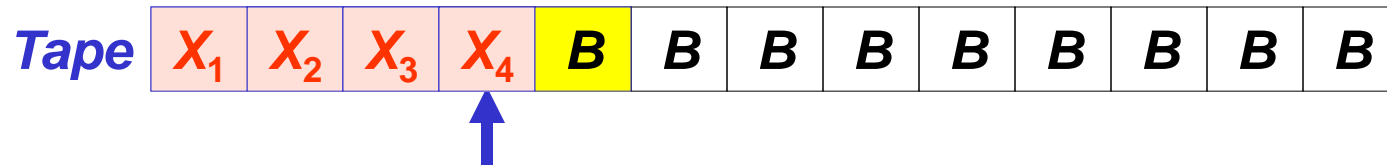


## *Stack machine*

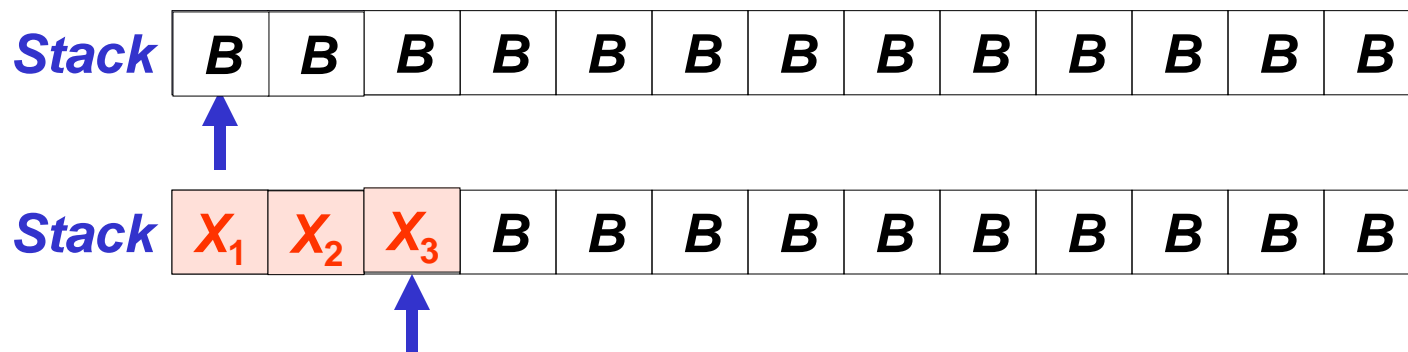


# Stack Machine

## *Turing machine*

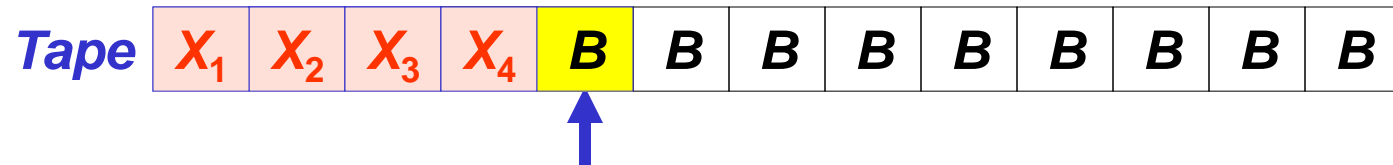


## *Stack machine*

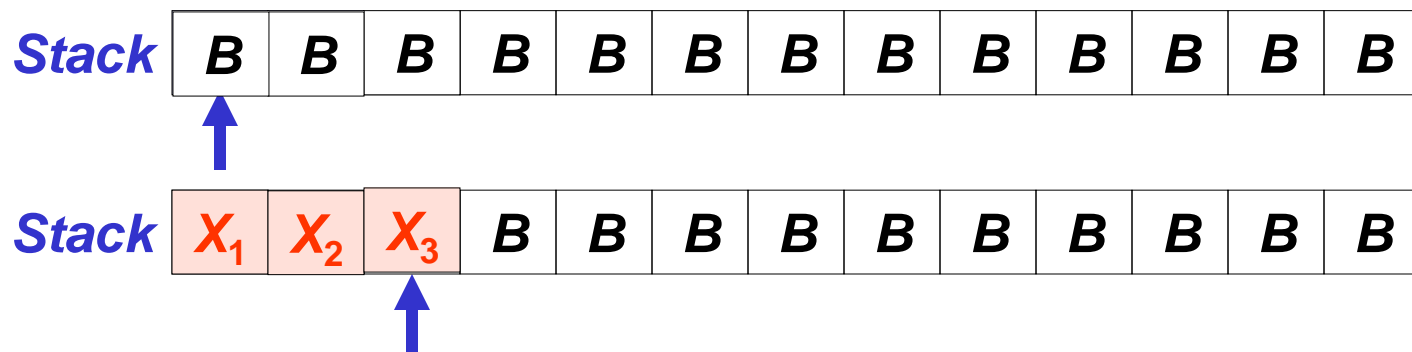


# Stack Machine

## *Turing machine*

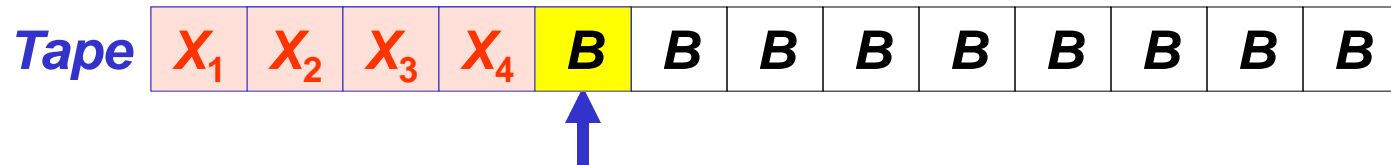


## *Stack machine*

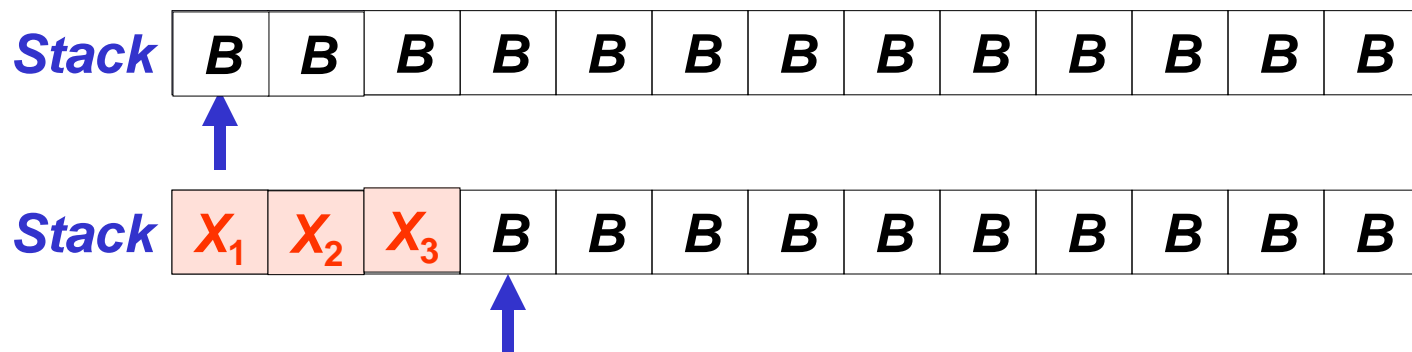


# Stack Machine

## *Turing machine*

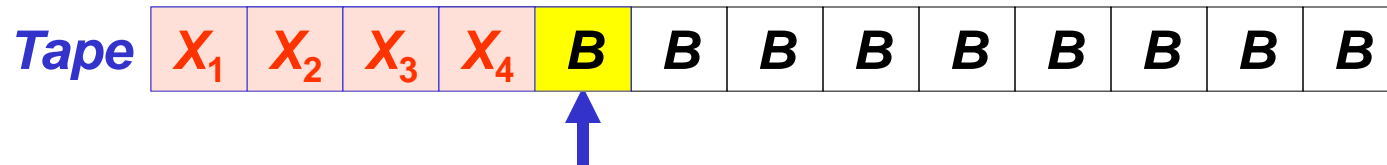


## *Stack machine*

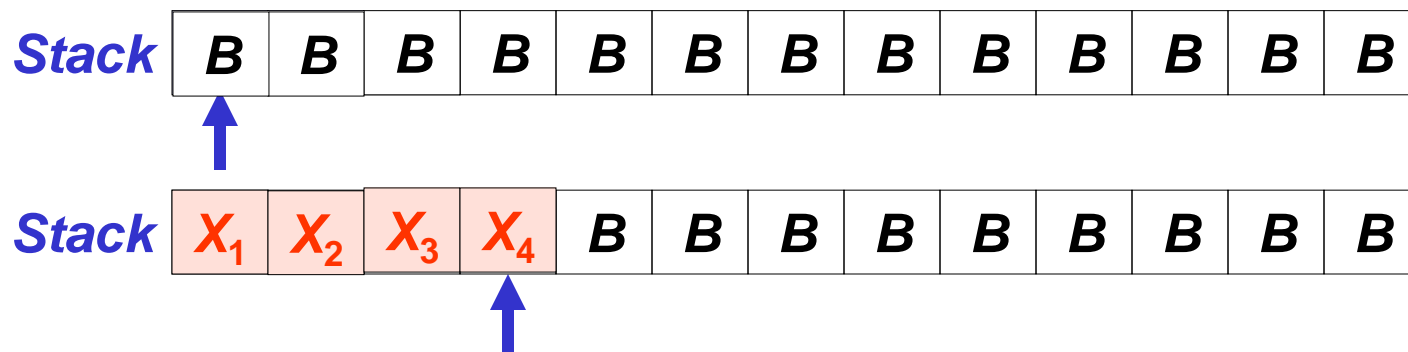


# Stack Machine

## Turing machine

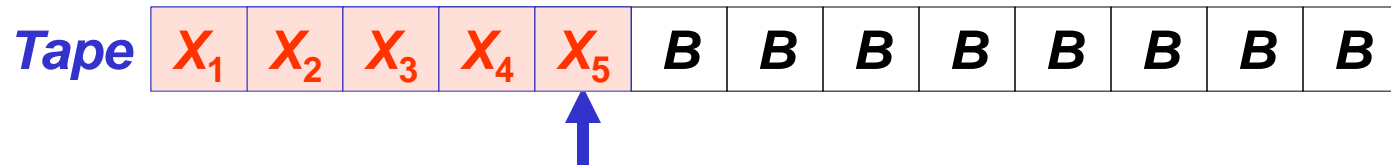


## Stack machine

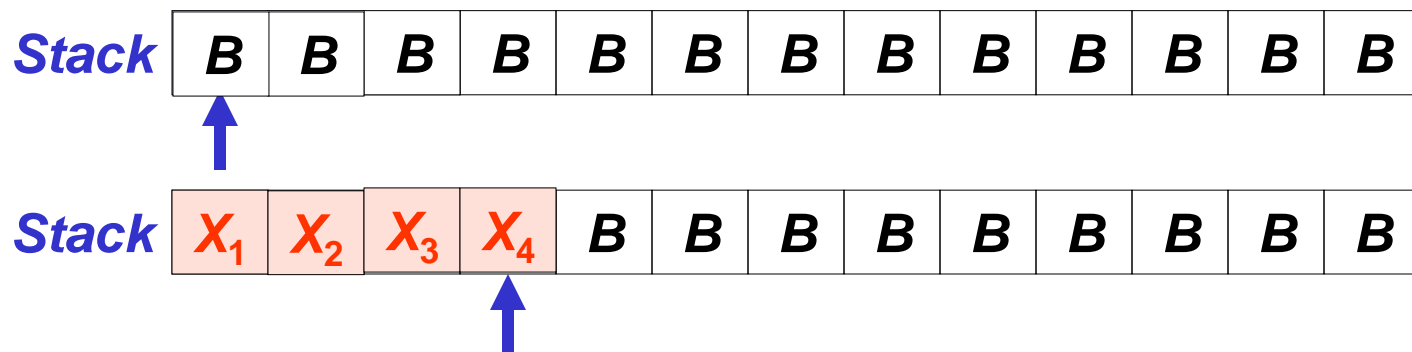


# Stack Machine

## *Turing machine*

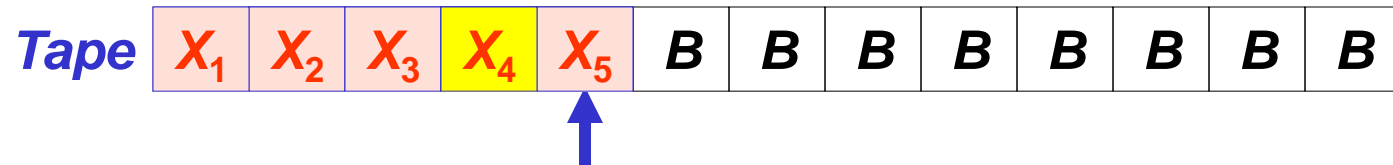


## *Stack machine*

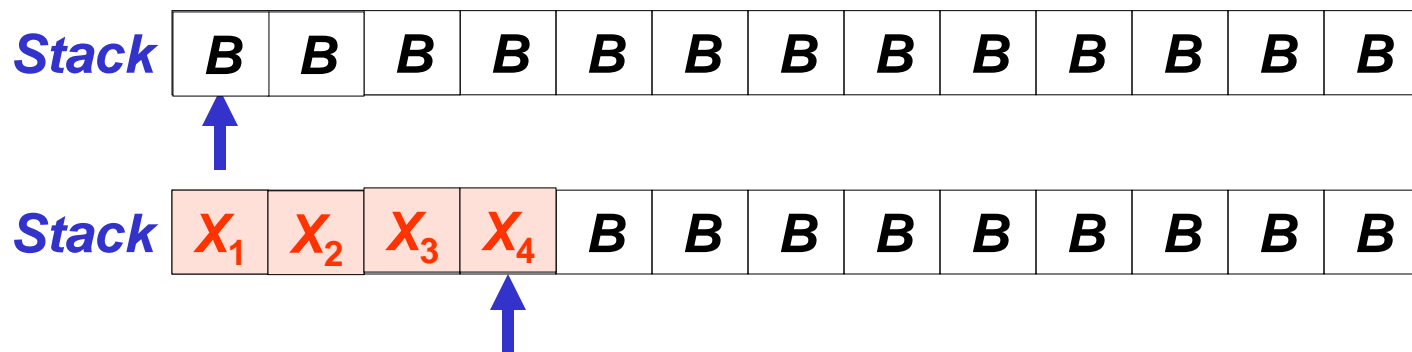


# Stack Machine

## *Turing machine*

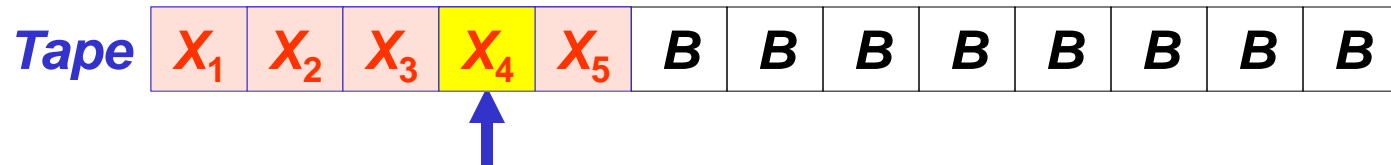


## *Stack machine*

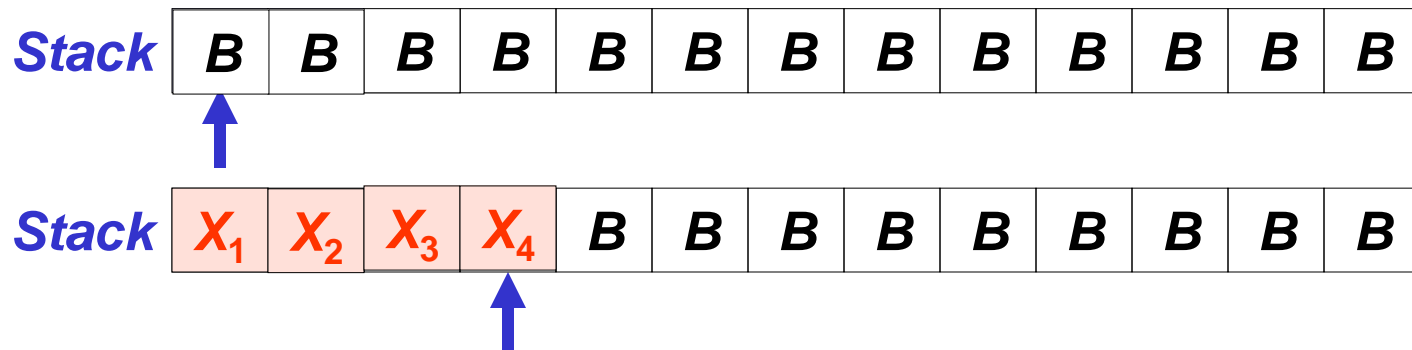


# Stack Machine

## *Turing machine*



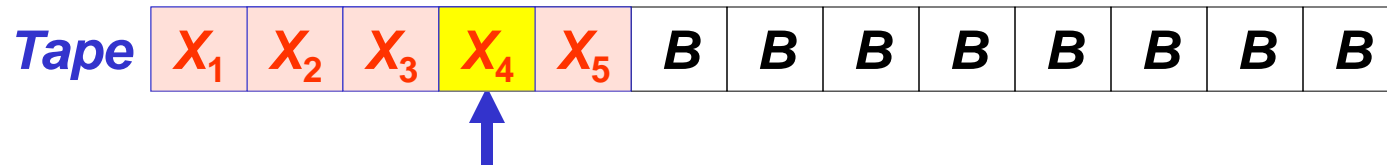
## *Stack machine*



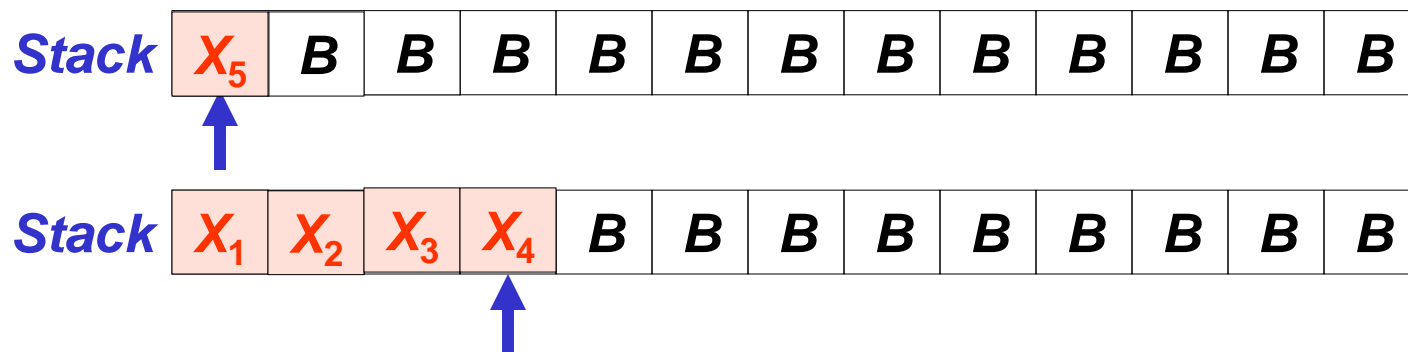


# Stack Machine

## *Turing machine*

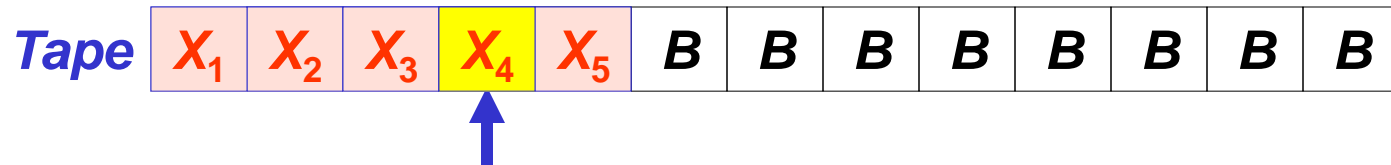


## *Stack machine*

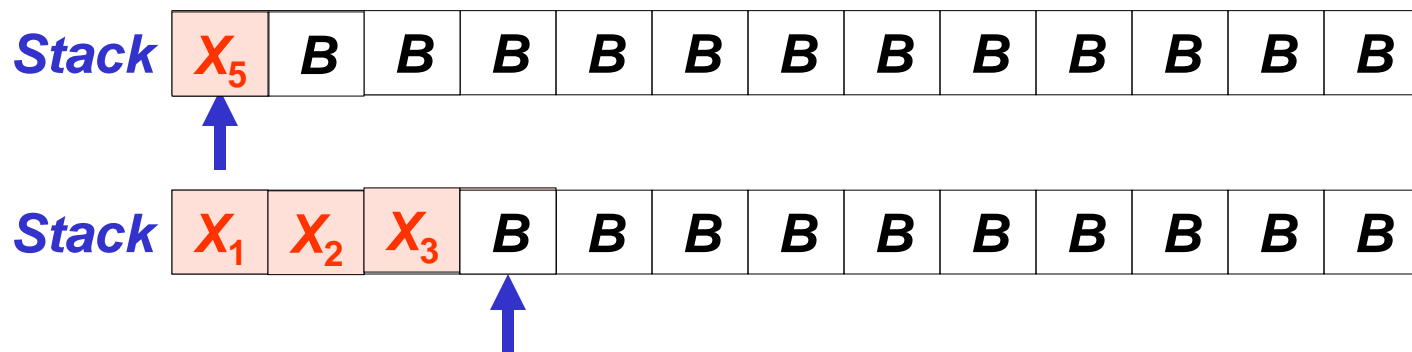


# Stack Machine

## *Turing machine*

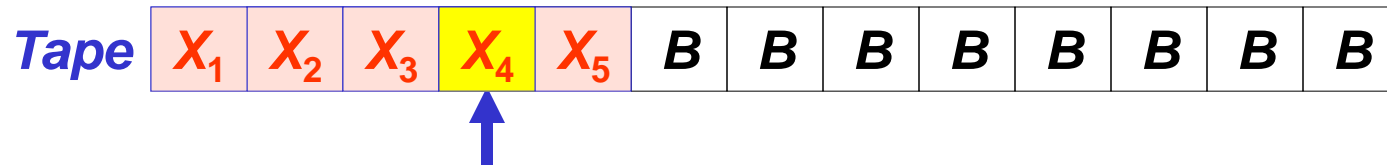


## *Stack machine*

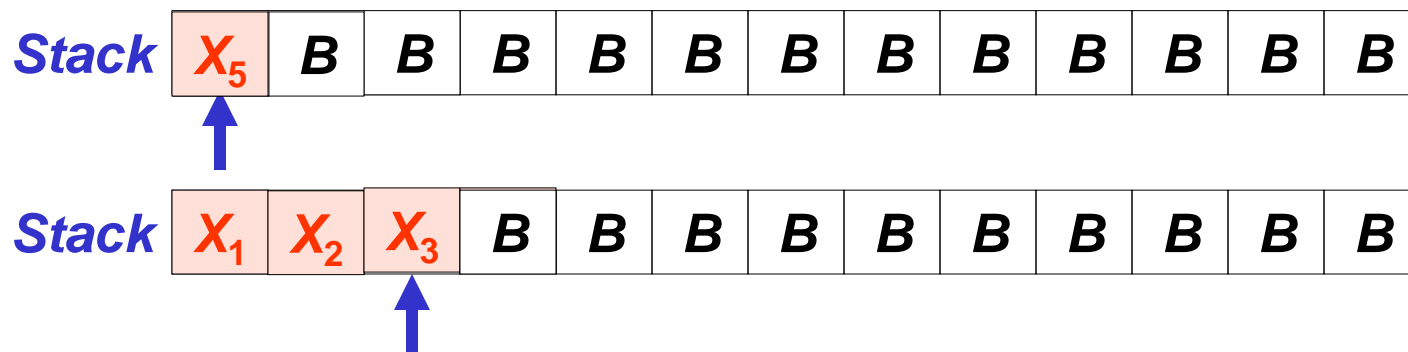


# Stack Machine

## *Turing machine*

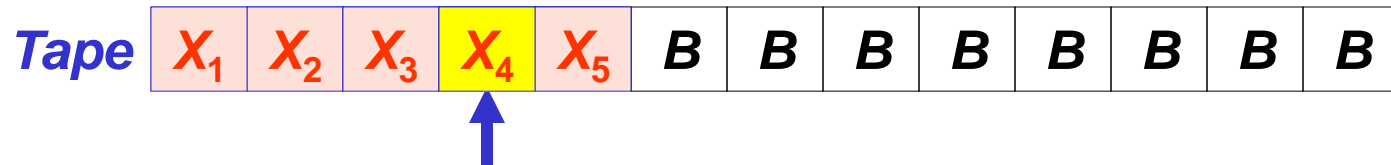


## *Stack machine*

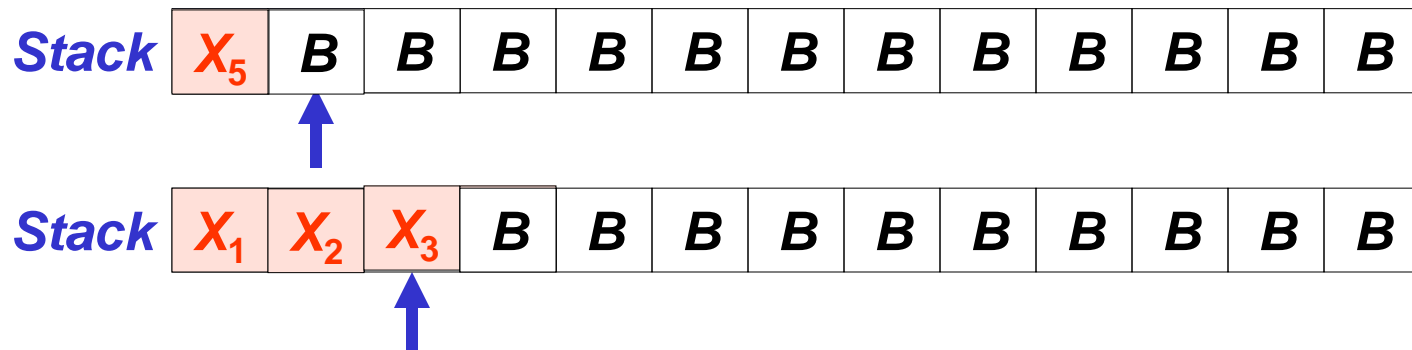


# Stack Machine

## *Turing machine*

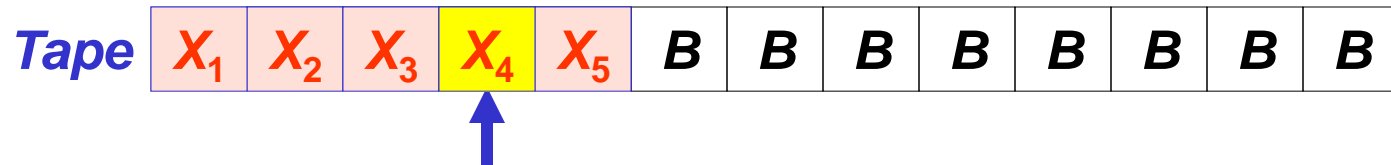


## *Stack machine*

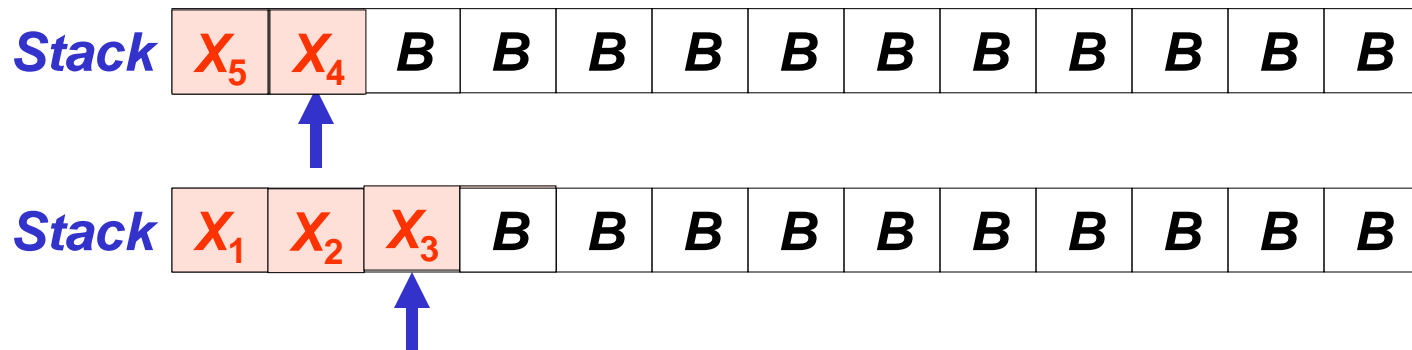


# Stack Machine

## *Turing machine*

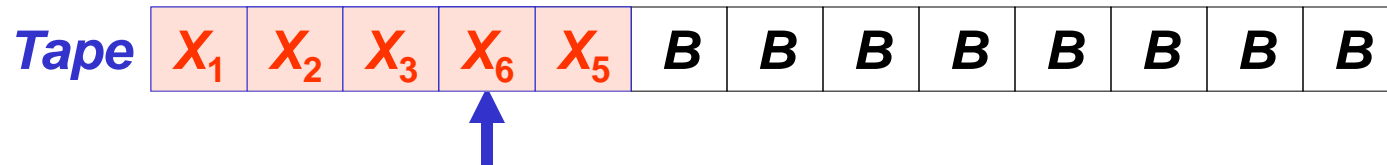


## *Stack machine*

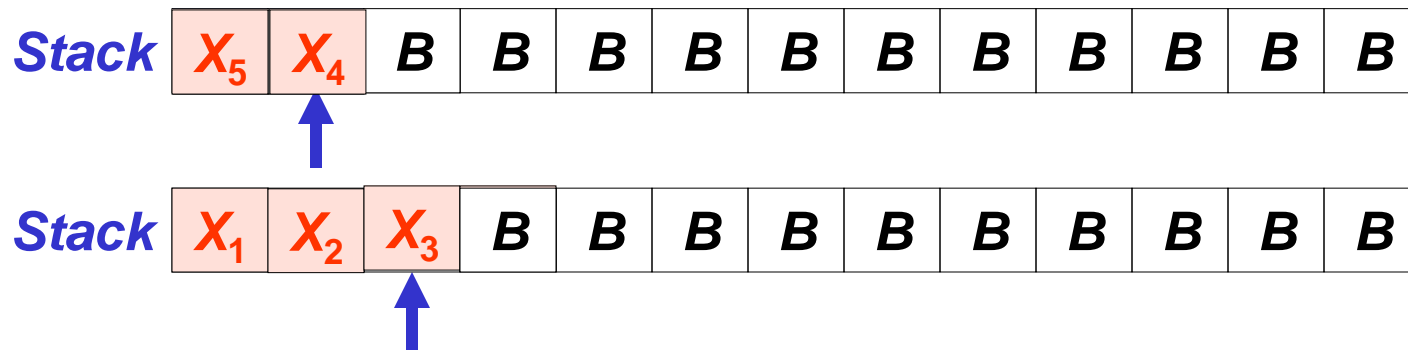


# Stack Machine

## Turing machine

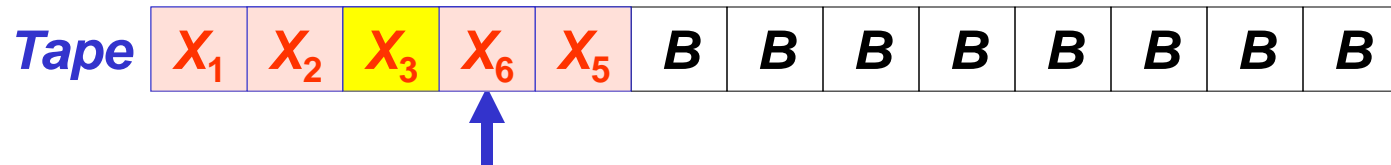


## Stack machine

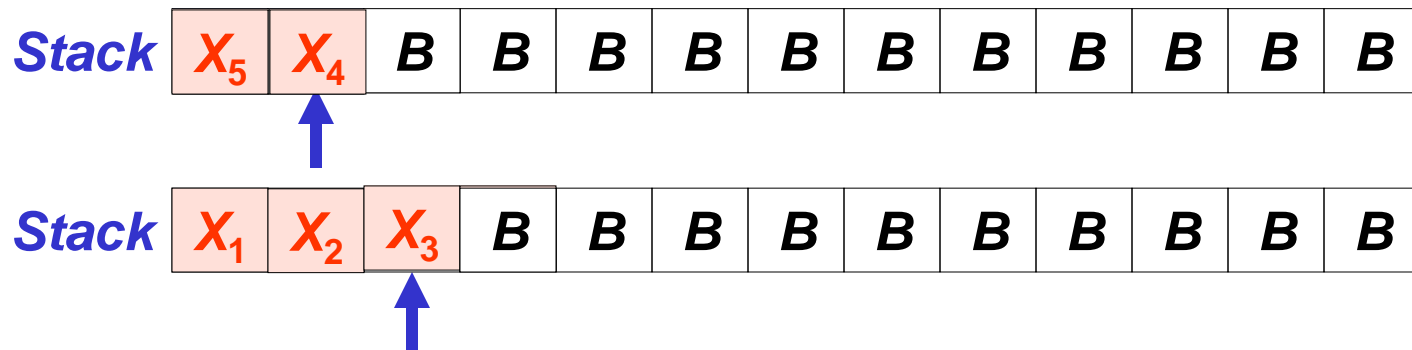


# Stack Machine

## Turing machine

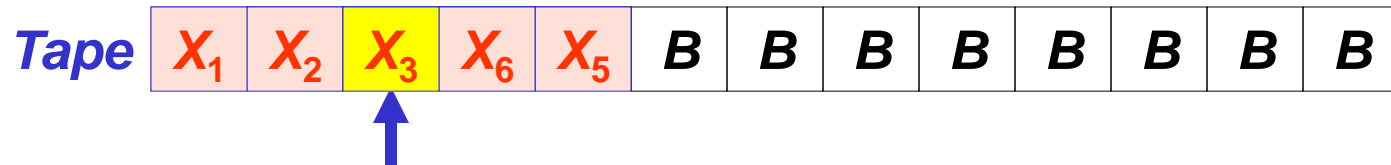


## Stack machine

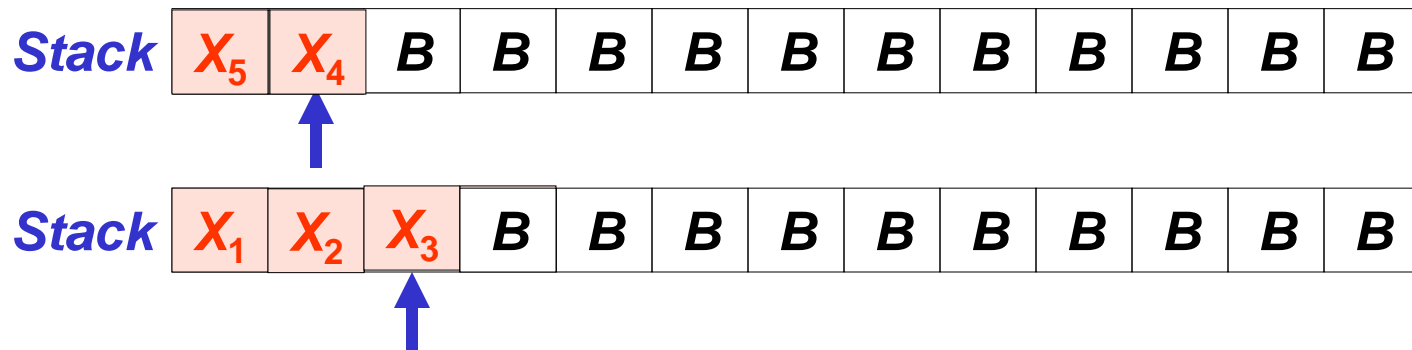


# Stack Machine

## Turing machine



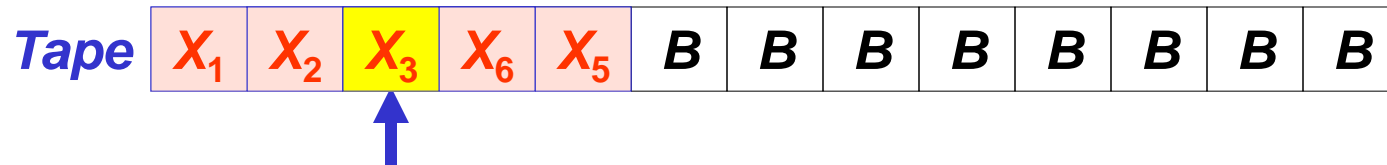
## Stack machine



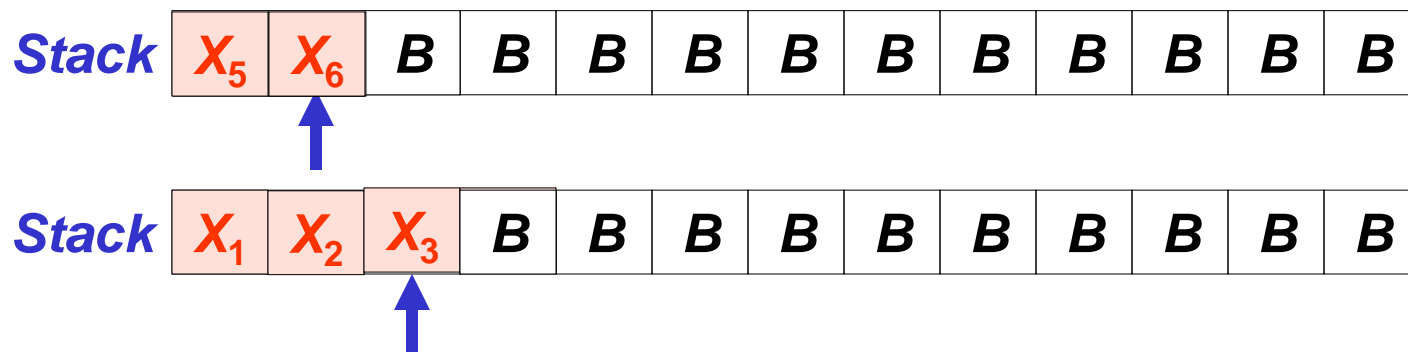


# Stack Machine

## *Turing machine*

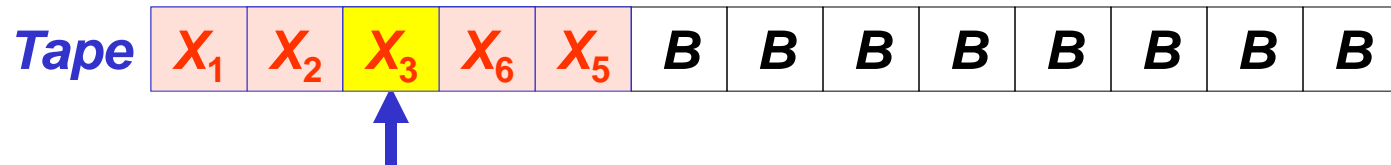


## *Stack machine*

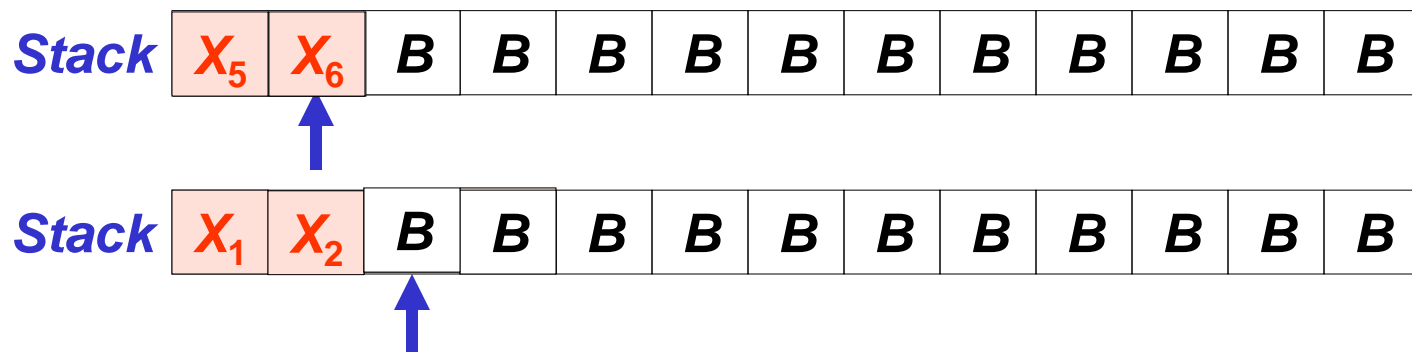


# Stack Machine

## *Turing machine*

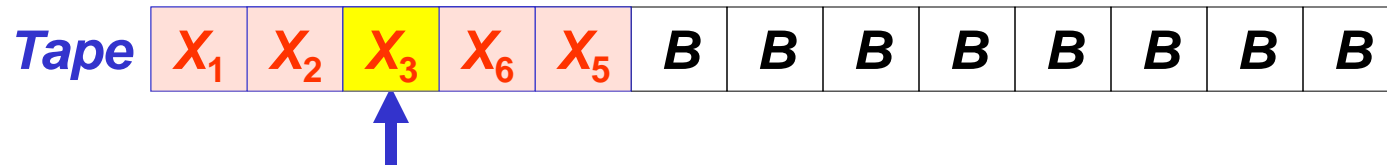


## *Stack machine*

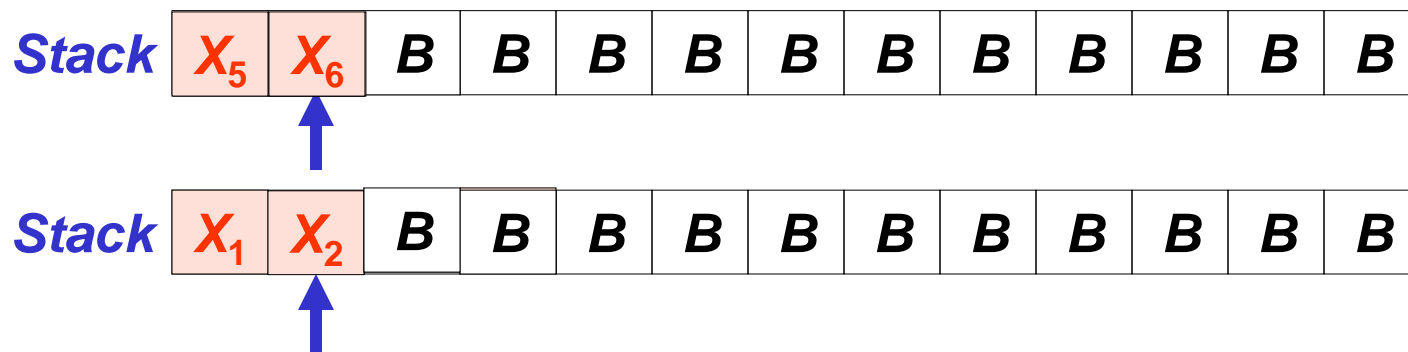


# Stack Machine

## *Turing machine*

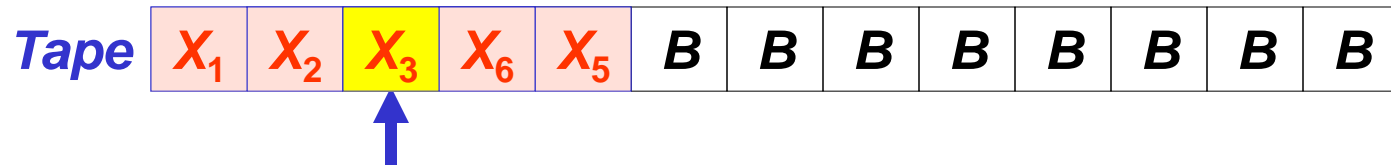


## *Stack machine*

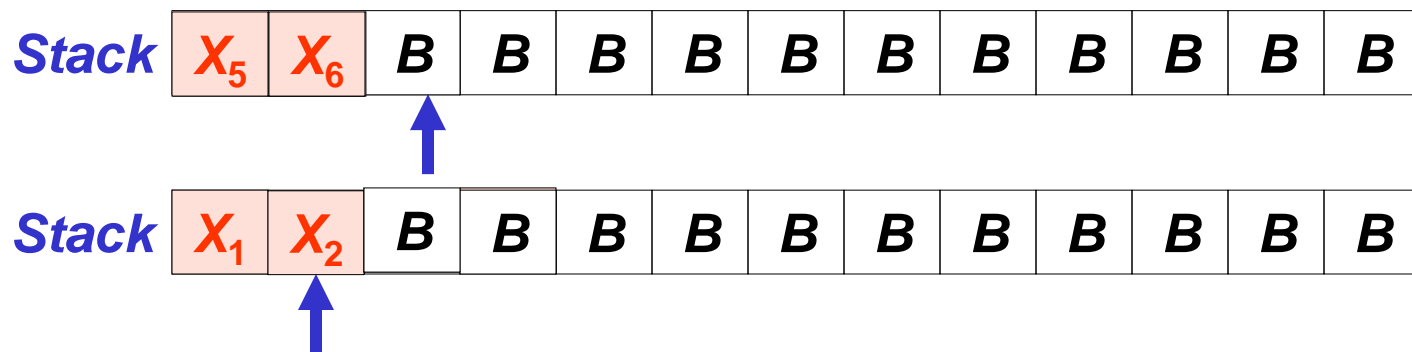


# Stack Machine

## *Turing machine*

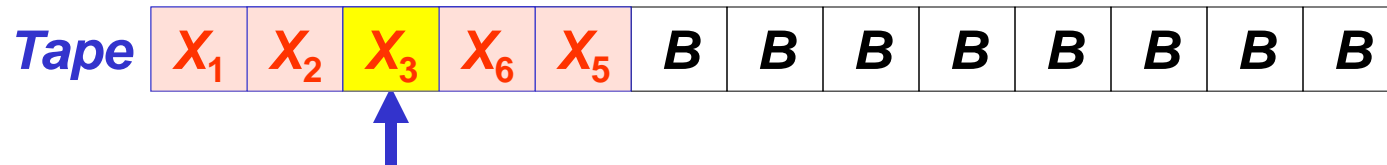


## *Stack machine*

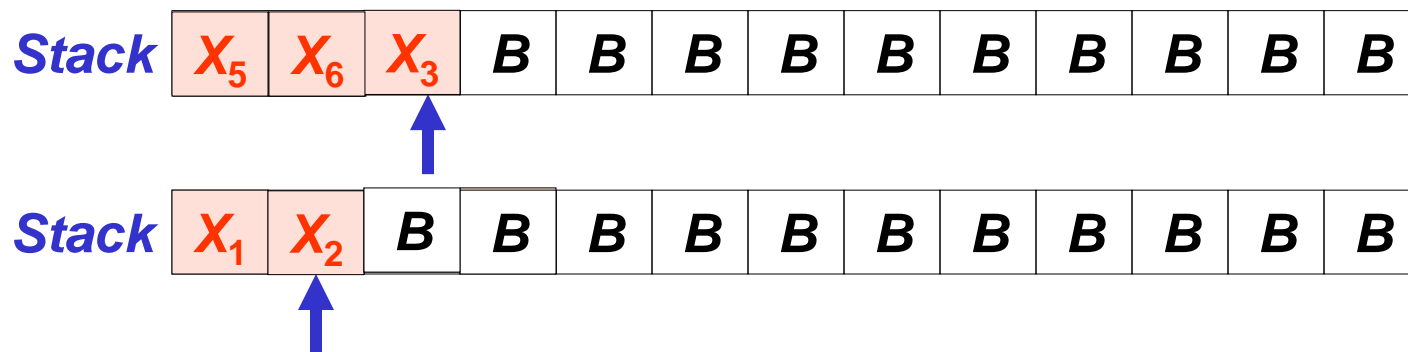


# Stack Machine

## *Turing machine*

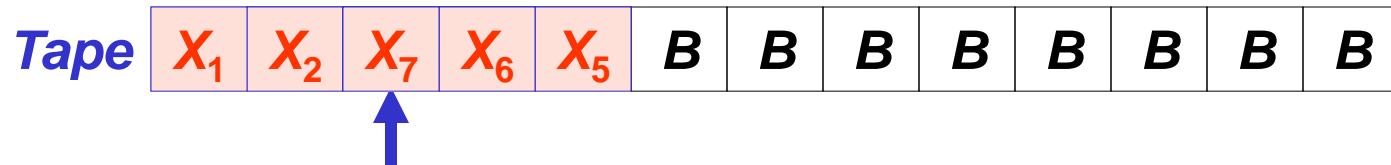


## *Stack machine*

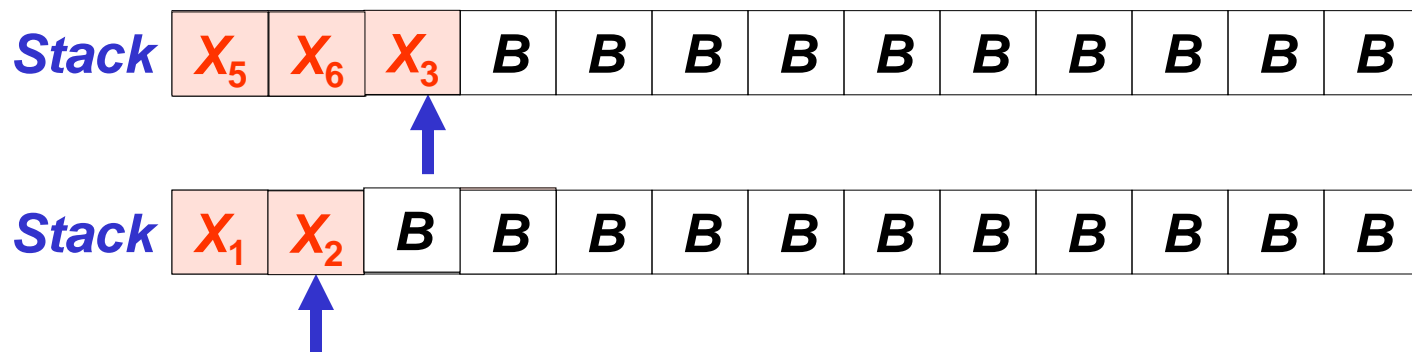


# Stack Machine

## *Turing machine*

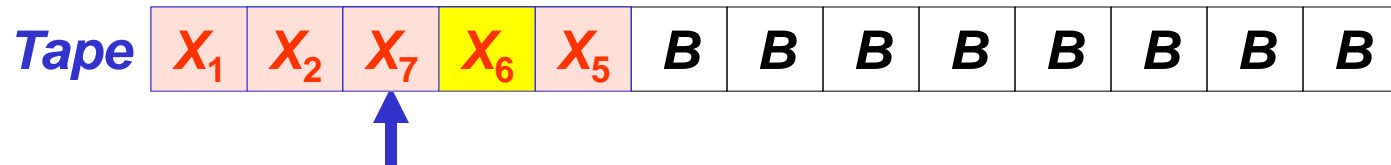


## *Stack machine*

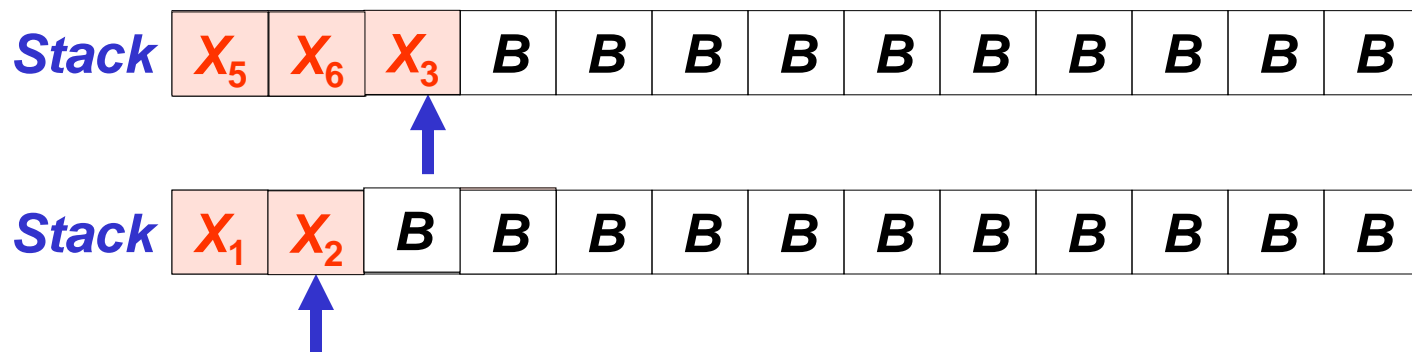


# Stack Machine

## Turing machine

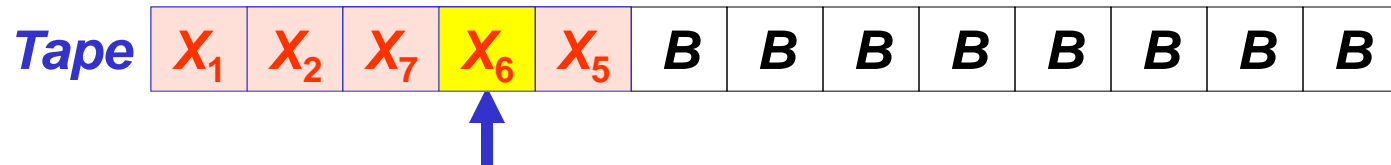


## Stack machine

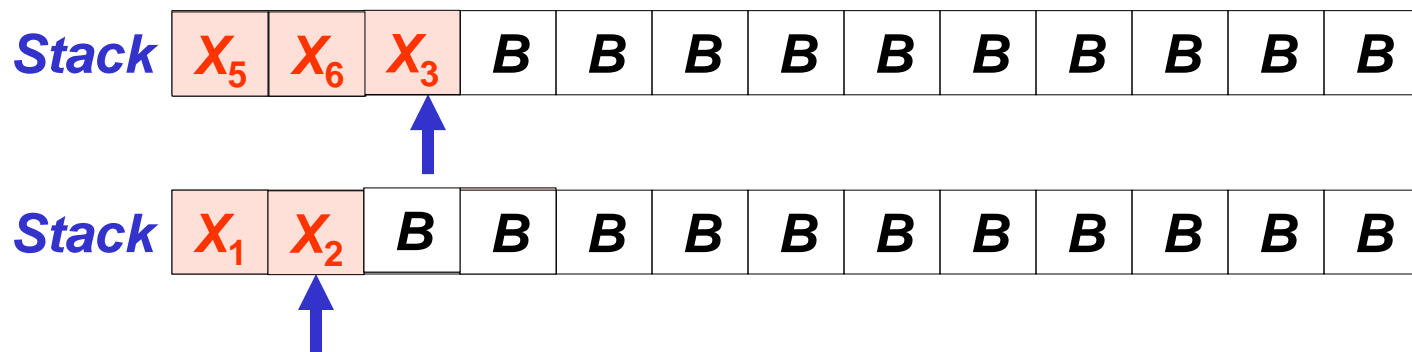


# Stack Machine

## Turing machine



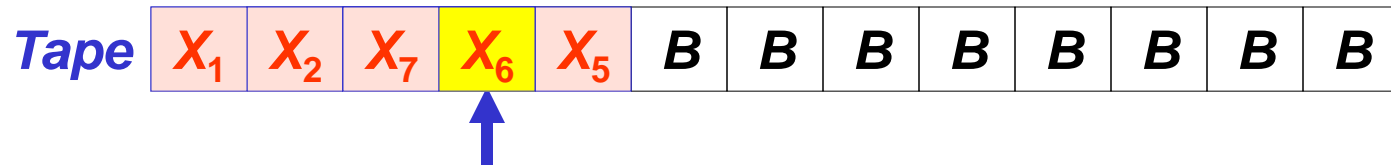
## Stack machine



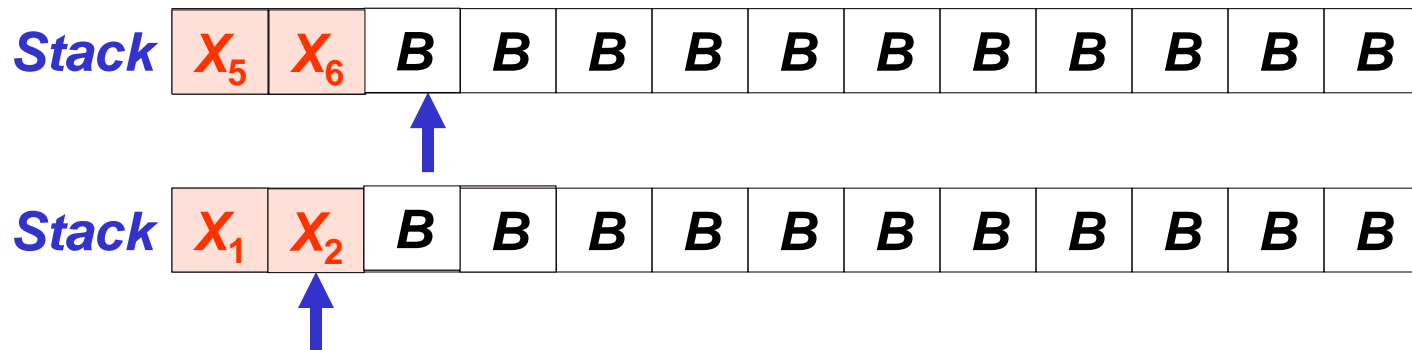


# Stack Machine

## *Turing machine*

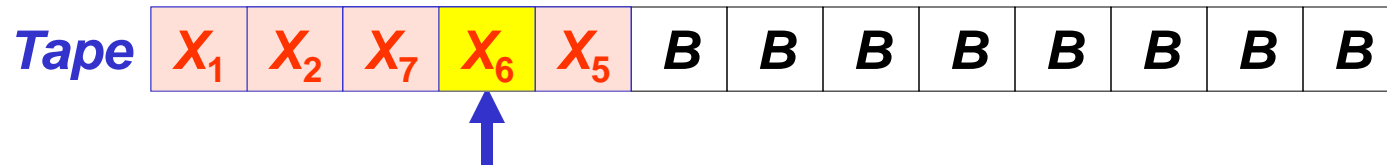


## *Stack machine*

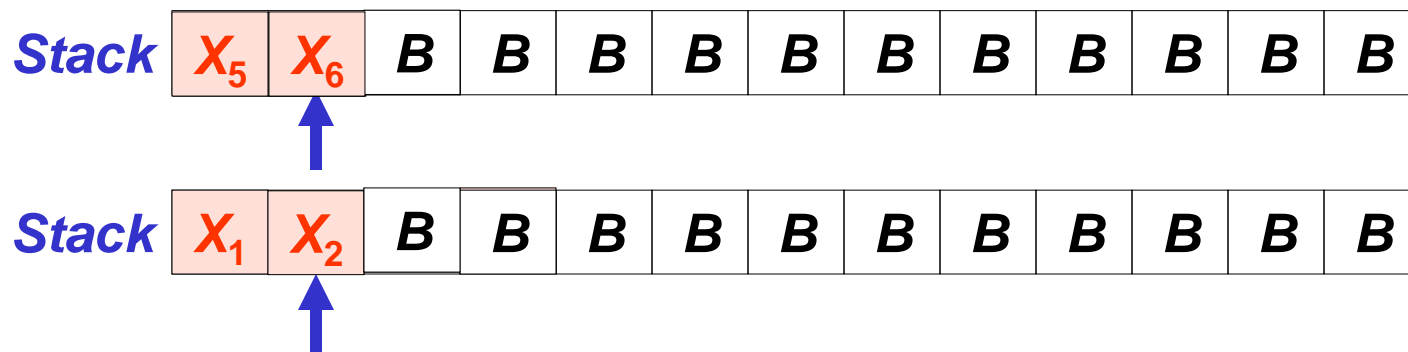


# Stack Machine

## *Turing machine*

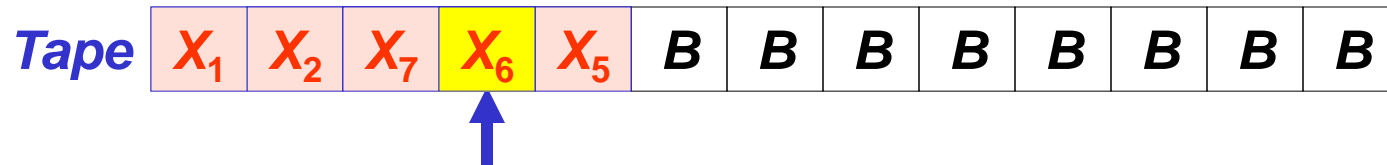


## *Stack machine*

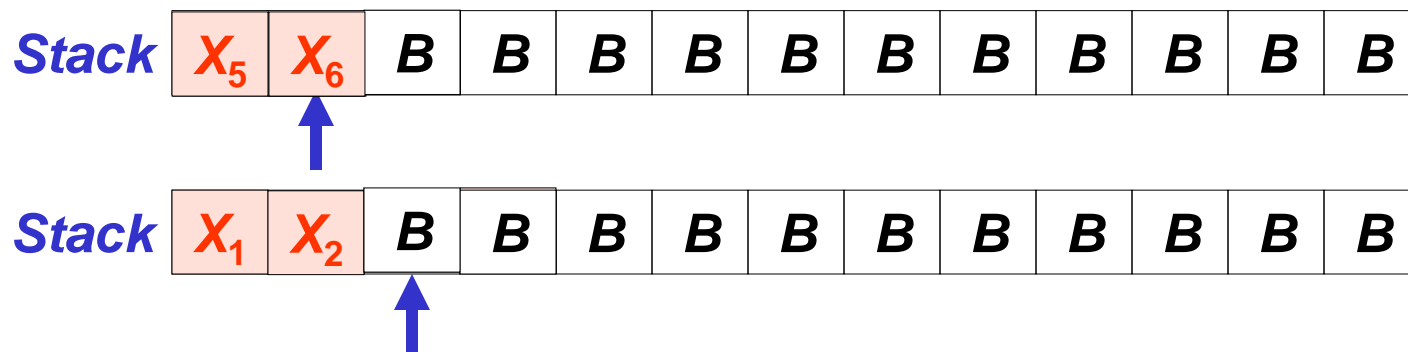


# Stack Machine

## *Turing machine*

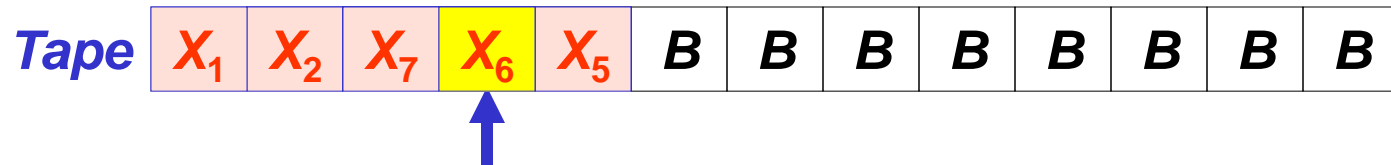


## *Stack machine*

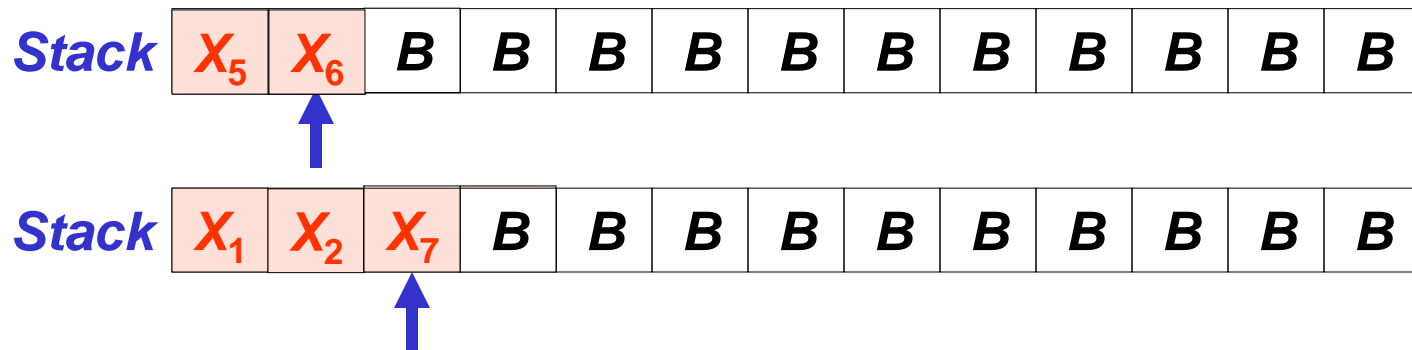


# Stack Machine

## *Turing machine*

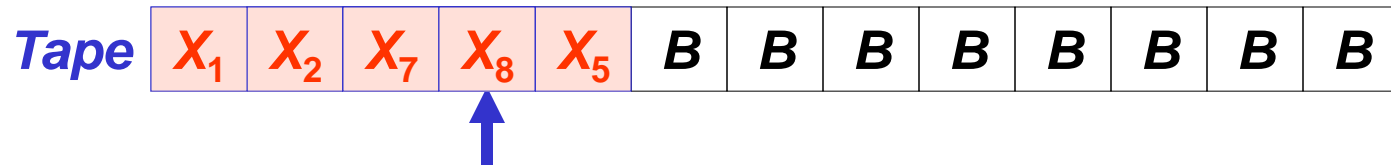


## *Stack machine*

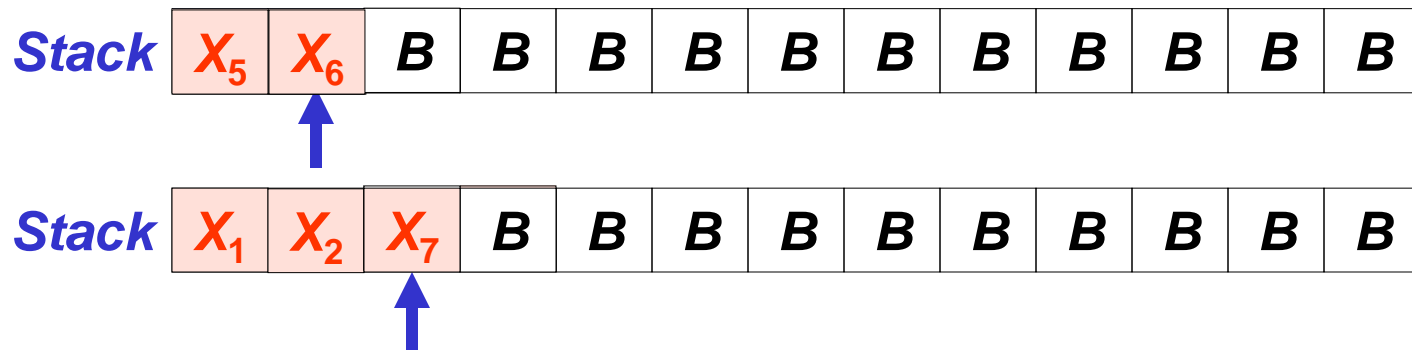


# Stack Machine

## Turing machine

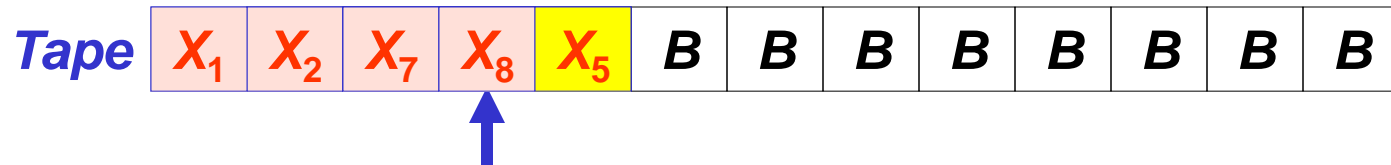


## Stack machine

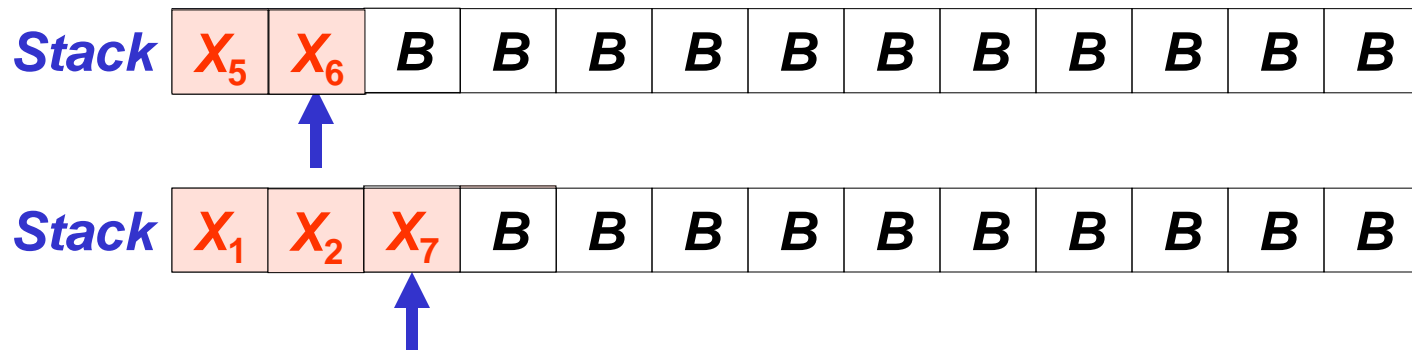


# Stack Machine

## *Turing machine*

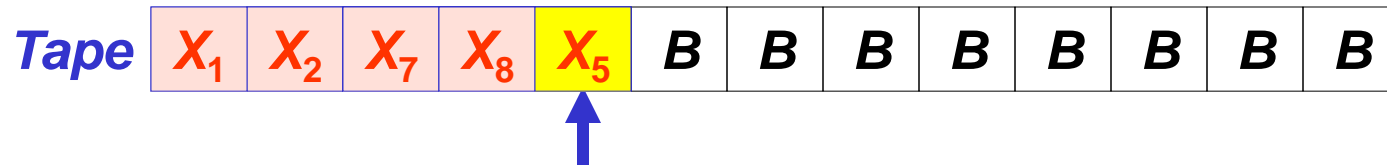


## *Stack machine*

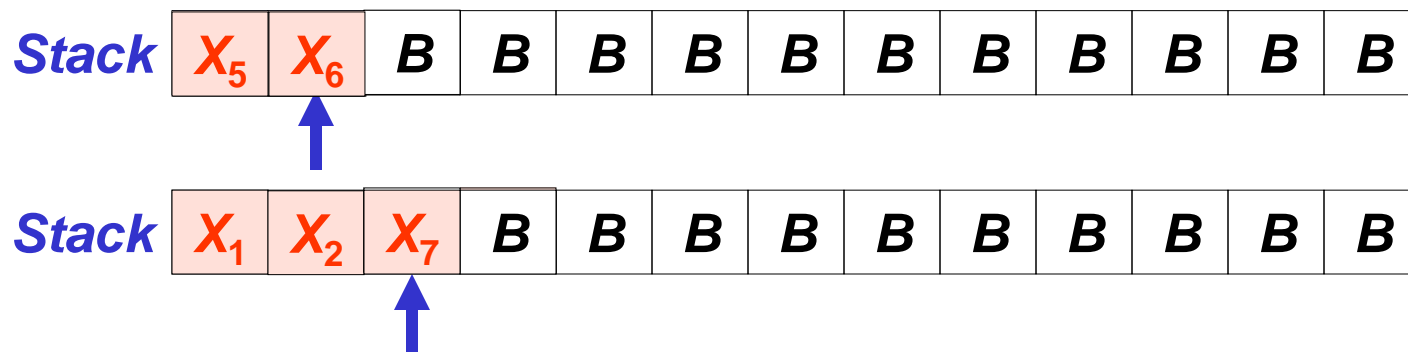


# Stack Machine

## *Turing machine*

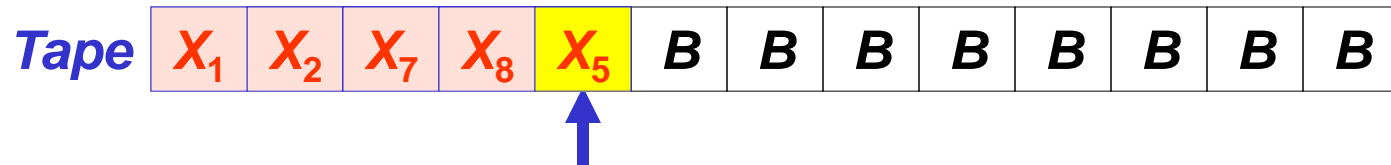


## *Stack machine*

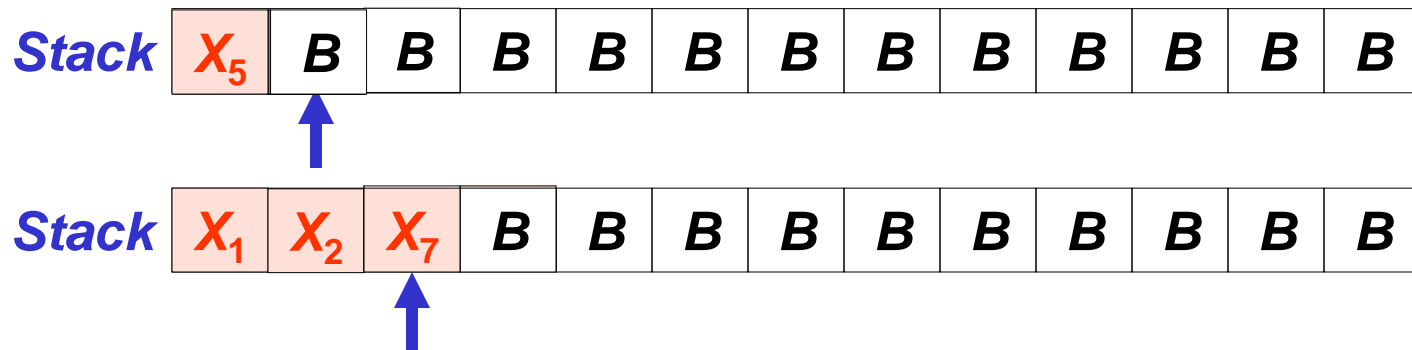


# Stack Machine

## Turing machine



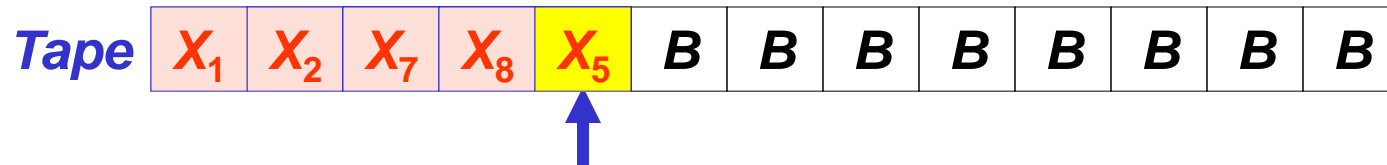
## Stack machine



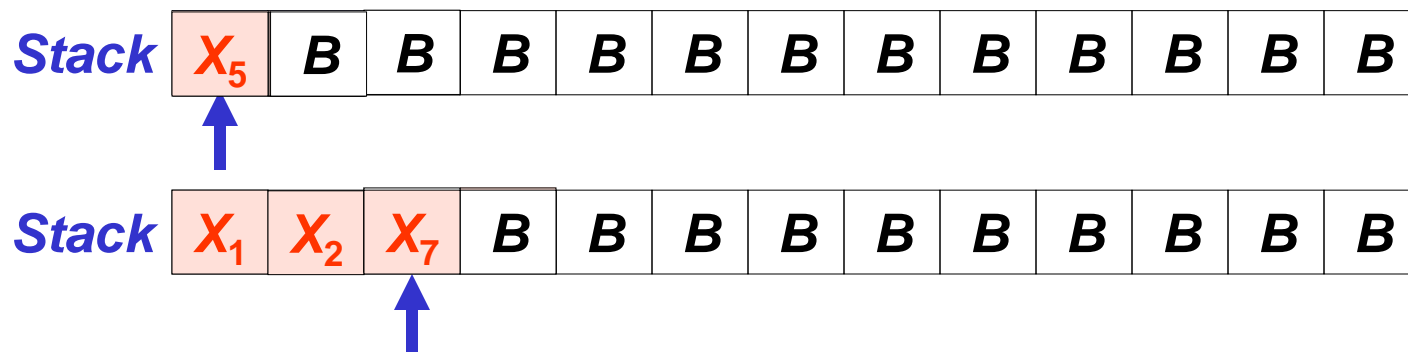


# Stack Machine

## Turing machine

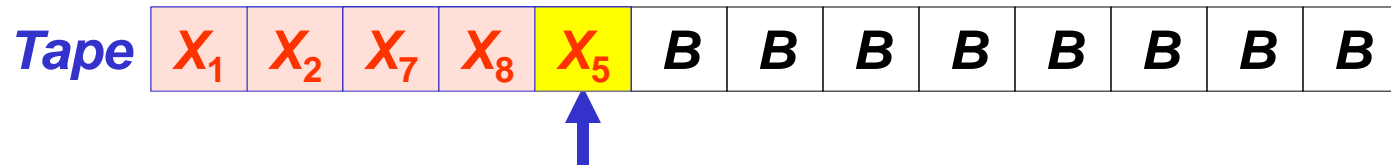


## Stack machine

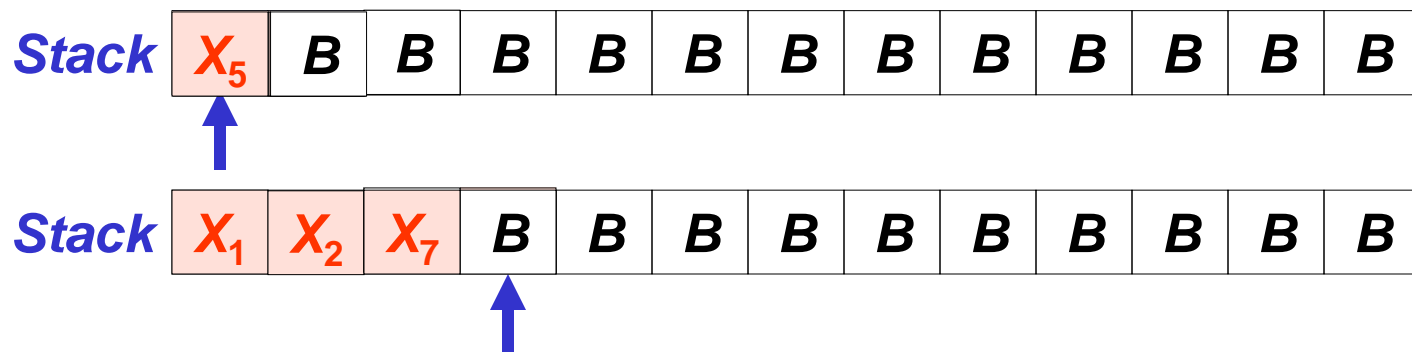


# Stack Machine

## *Turing machine*

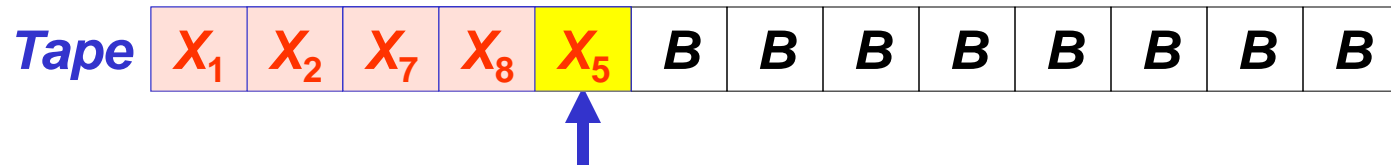


## *Stack machine*

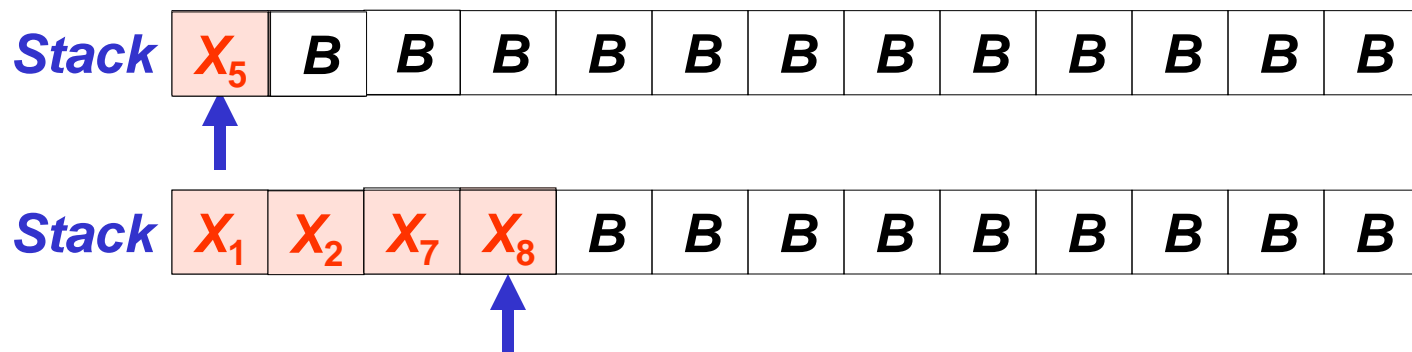


# Stack Machine

## *Turing machine*



## *Stack machine*



# Counter Machine

# Counter Machine

***Stack machine***

# Counter Machine

## Stack machine

*Input tape*

¢	$a_1$	$a_2$	$a_3$	\$
---	-------	-------	-------	----

*Stack*

$a_3$	$a_2$	$a_1$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-------	-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

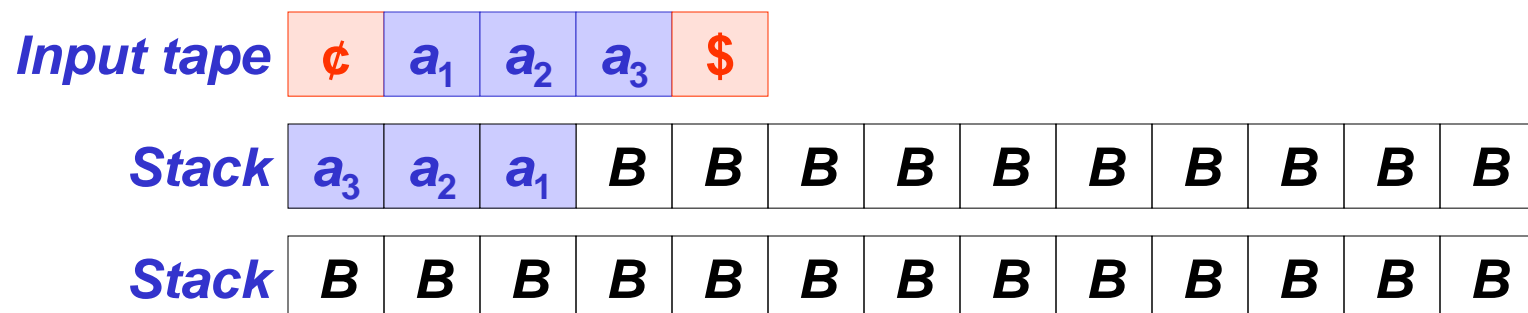
*Stack*

$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

---

# Counter Machine

## Stack machine



## Counter machine

# Counter Machine

## Stack machine

<i>Input tape</i>	¢	$a_1$	$a_2$	$a_3$	\$								
<i>Stack</i>	$a_3$	$a_2$	$a_1$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
<i>Stack</i>	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$

## Counter machine

<i>Counter</i>	X	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
<i>Counter</i>	X	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
<i>Counter</i>	X	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
<i>Counter</i>	X	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$



# Counter Machine

# Counter Machine

***Stack machine***

# Counter Machine

## Stack machine



# Counter Machine

## Stack machine



---

## Counter Machine

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	--	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	----	----------	-----	-----	-----	-----	-----	-----	-----

## Counter Machine

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

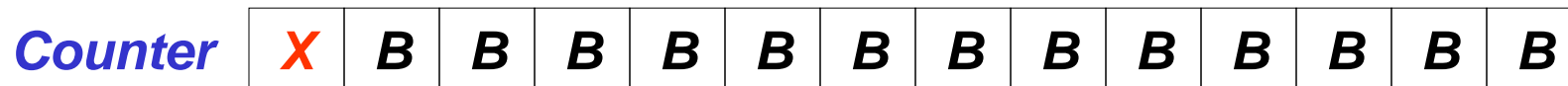
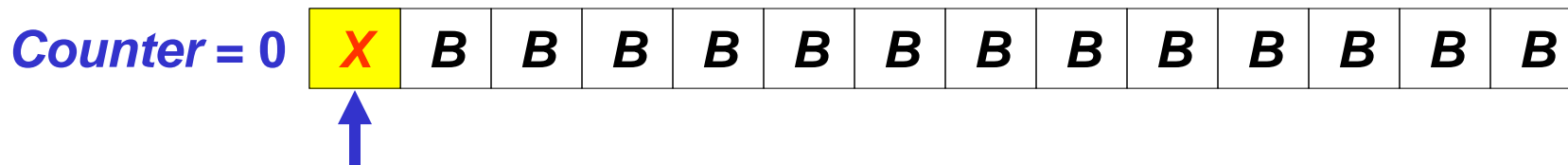
Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine



## Counter Machine

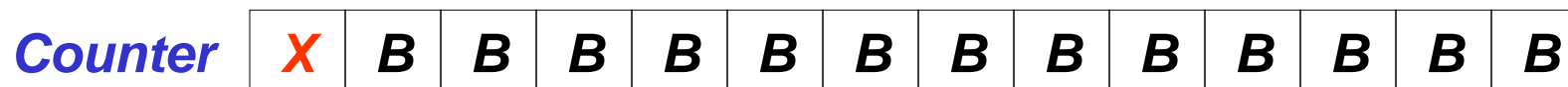
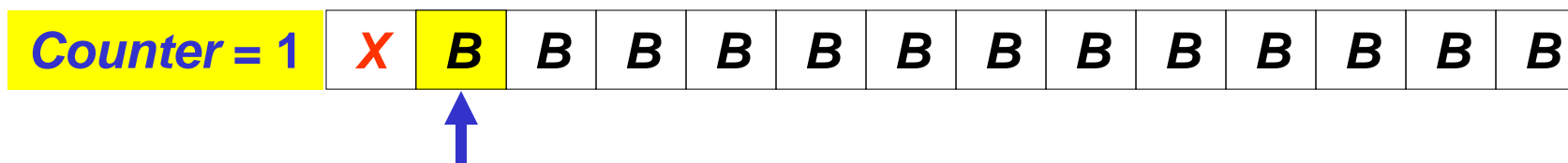


# Counter Machine

## Stack machine



## Counter Machine

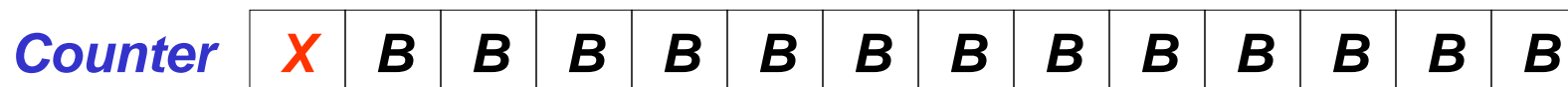
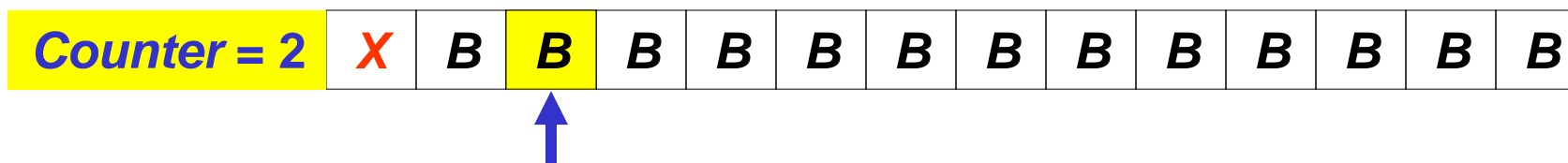


# Counter Machine

## Stack machine



## Counter Machine



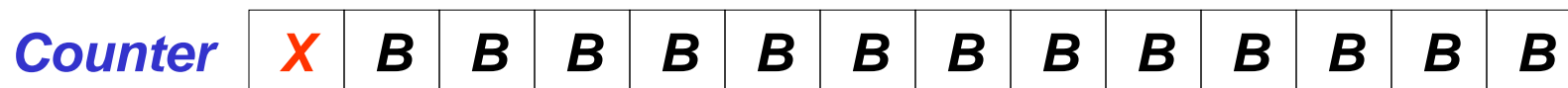
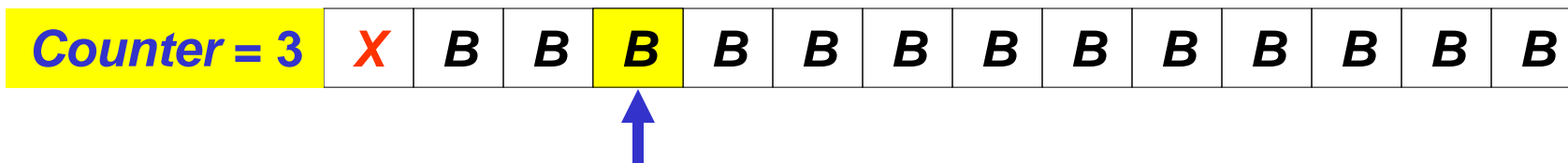


# Counter Machine

## Stack machine



## Counter Machine

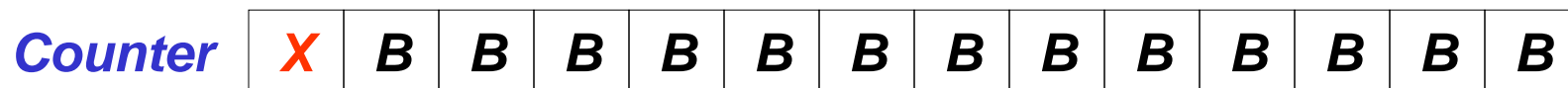
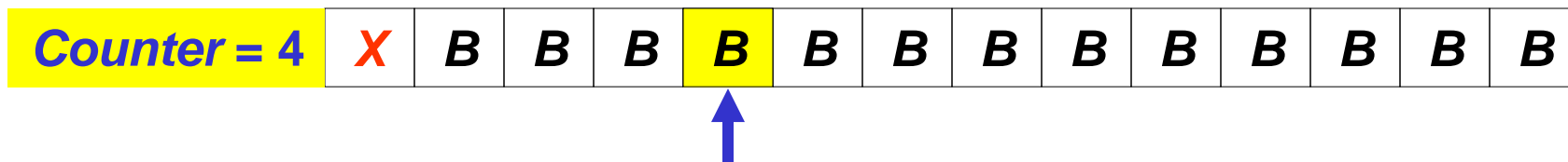


# Counter Machine

## Stack machine



## Counter Machine

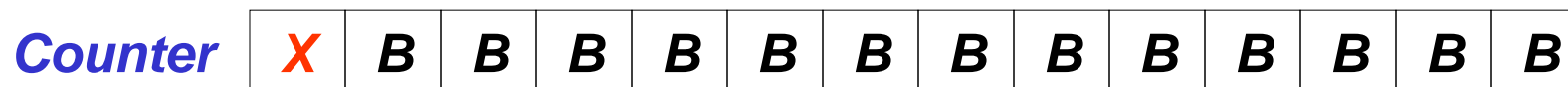
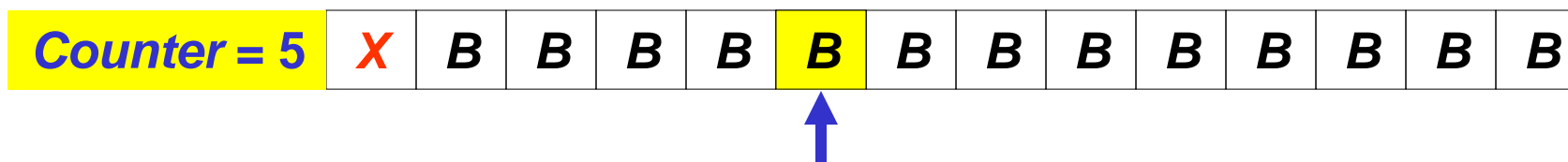


# Counter Machine

## Stack machine



## Counter Machine

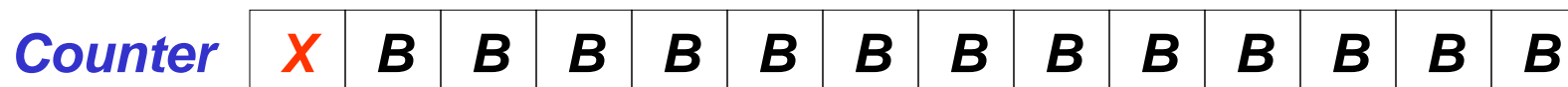
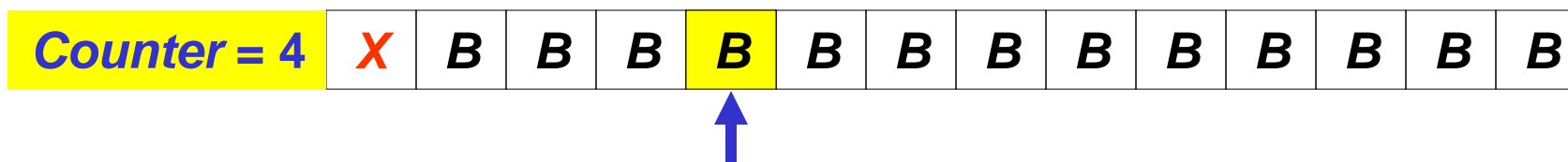


# Counter Machine

## Stack machine



## Counter Machine

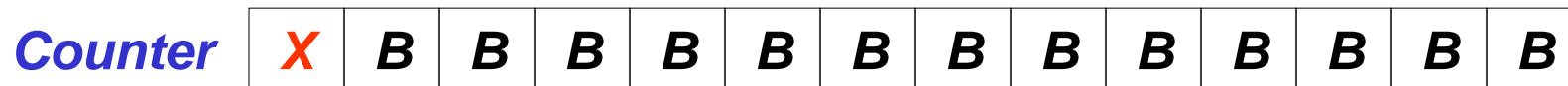
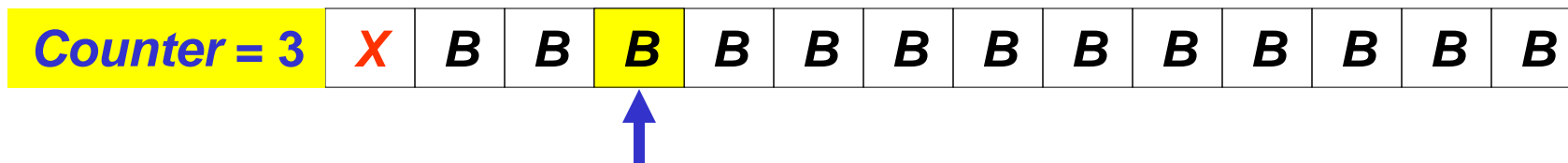


# Counter Machine

## Stack machine



## Counter Machine

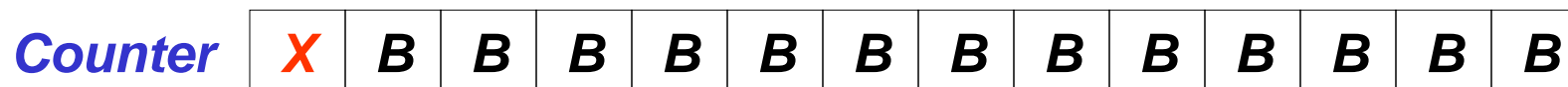
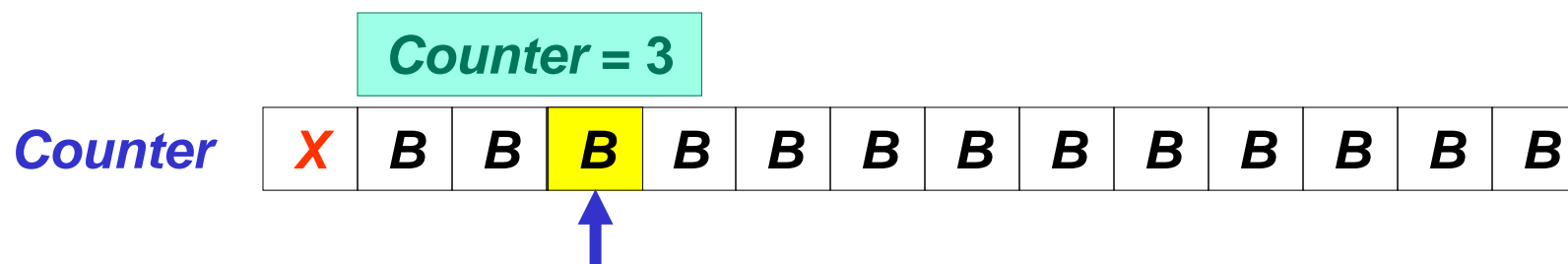


# Counter Machine

## Stack machine



## Counter machine

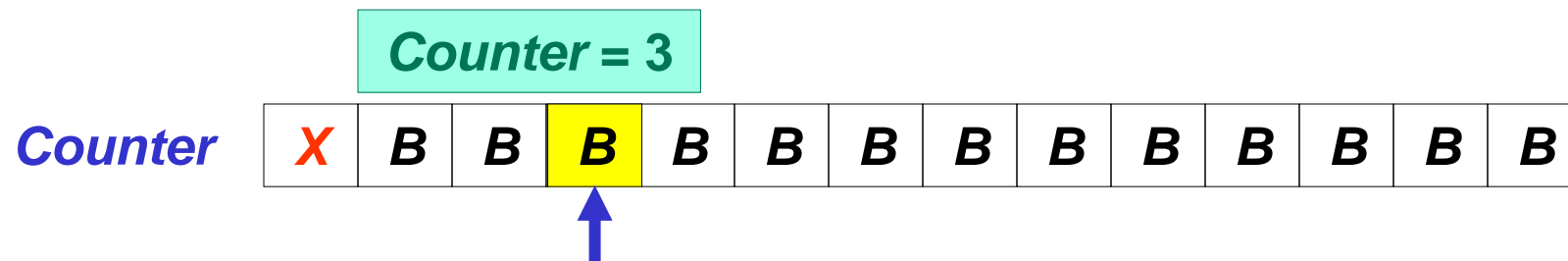


# Counter Machine

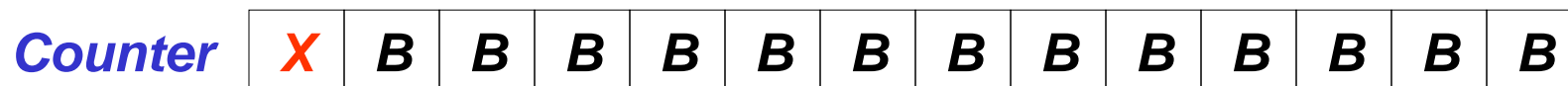
## Stack machine



## Counter machine



[  $q$ ,  $p$ , Counter to  $k$  ]

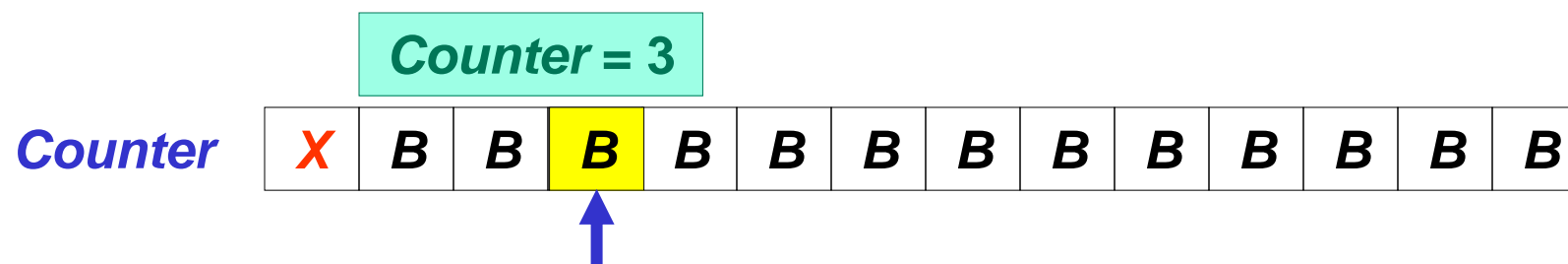


# Counter Machine

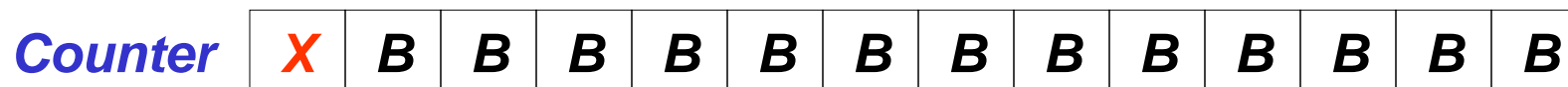
## Stack machine



## Counter machine



[  $q$ ,  $p$ , 4 ]



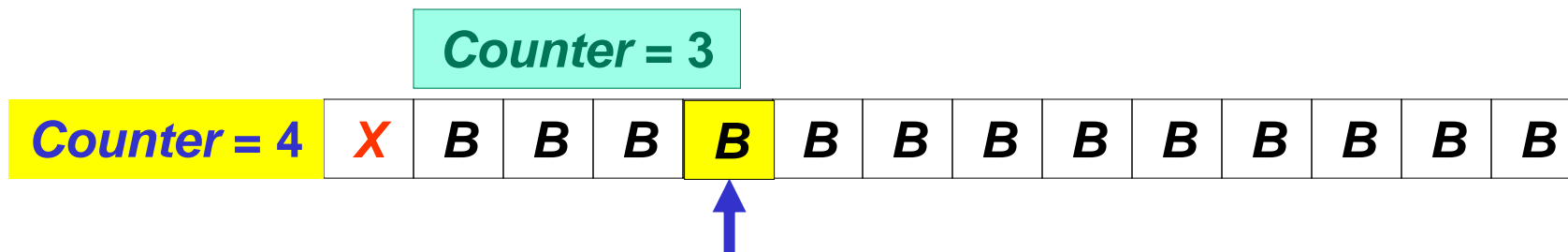


# Counter Machine

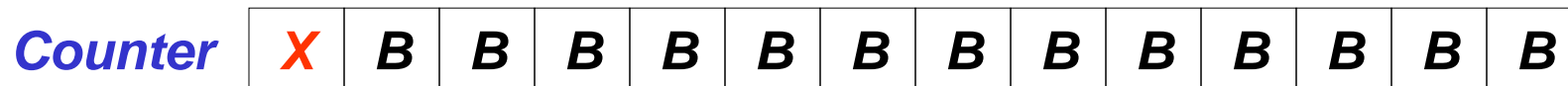
## Stack machine



## Counter machine



[  $q$ ,  $p$ , 3 ]

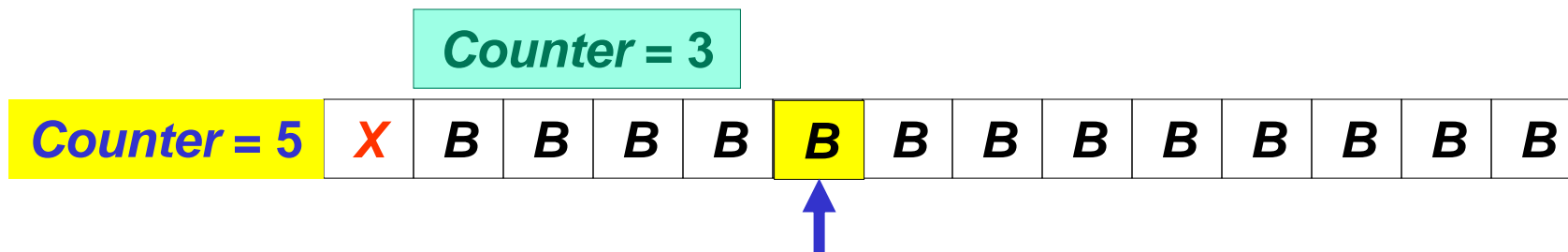


# Counter Machine

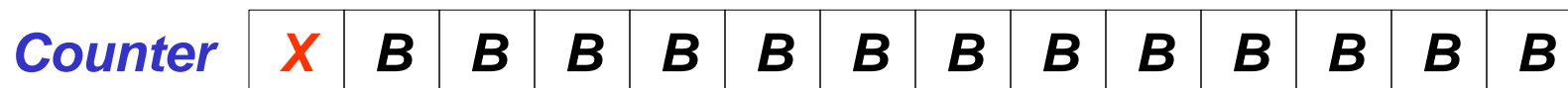
## Stack machine



## Counter machine



[  $q$ ,  $p$ , 2 ]

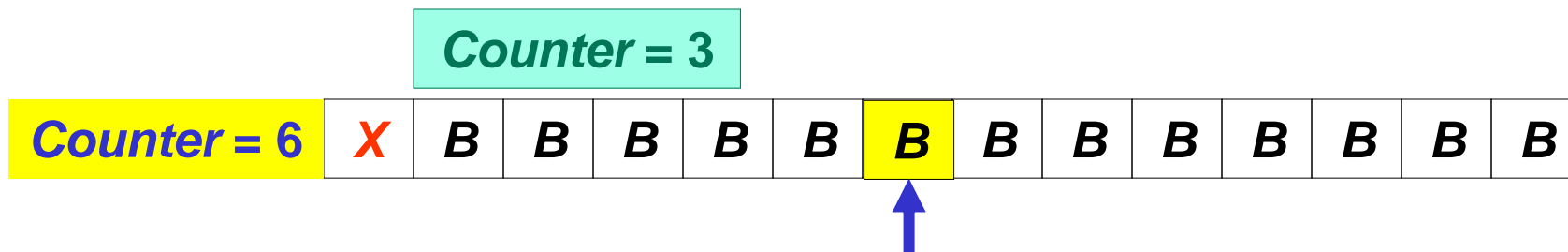


# Counter Machine

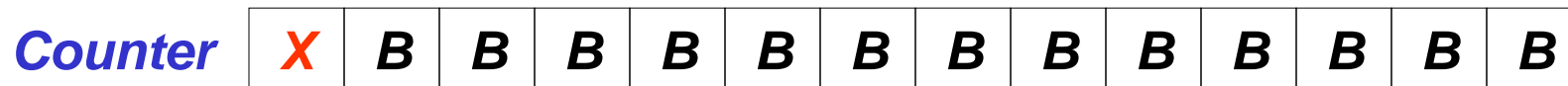
## Stack machine



## Counter machine



[  $q, p, 1$  ]

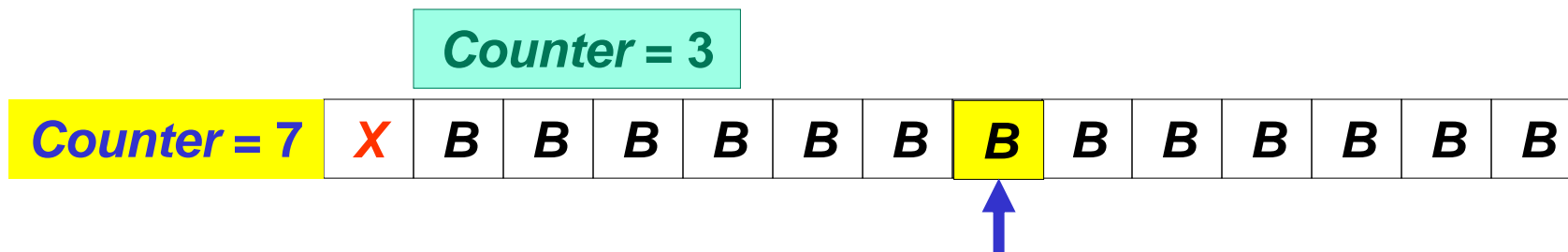


# Counter Machine

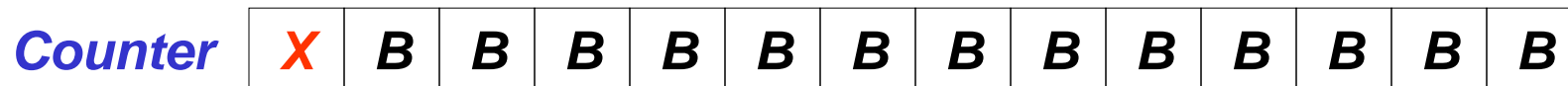
## Stack machine



## Counter machine



[  $q$ ,  $p$ , 0 ]



# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----	-----

## Counter machine

**$k$**  distinct stack symbols

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----	-----

## Counter machine

$k$  distinct stack symbols

$Z_0, Z_1, Z_2, \dots, Z_{k-1}$

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----	-----

## Counter machine

$k$  distinct stack symbols

$Z_0, Z_1, Z_2, \dots, Z_{k-1}$

symbol  $Z_i$  is coded as integer  $i$ ,  $0 \leq i \leq k-1$

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----



# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----

## Counter machine

$k$  distinct stack symbols

$Z_0, Z_1, Z_2, \dots, Z_{k-1}$

symbol  $Z_i$  is coded as integer  $i$ ,  $0 \leq i \leq k-1$

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----

## Counter machine

$k$  distinct stack symbols

$Z_0, Z_1, Z_2, \dots, Z_{k-1}$

symbol  $Z_i$  is coded as integer  $i$ ,  $0 \leq i \leq k-1$

Counter = $j$	$X$	$j = i_m + k i_{m-1} + k^2 i_{m-2} + k^3 i_{m-3} + \dots + k^{m-1} i_1$	$B$	$B$	$B$
---------------	-----	---	-----	-----	-----

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Putting symbol  $Z_r$  on the top of the stack

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Putting symbol  $Z_r$  on the top of the stack

Before: stack symbols' value =  $j$

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Putting symbol  $Z_r$  on the top of the stack

Before: stack symbols' value =  $j$   
 After: stack symbols' value =  $j k + r$

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Putting symbol  $Z_r$  on the top of the stack

Before: stack symbols' value =  $j$   
 After: stack symbols' value =  $j k + r$

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

[  $q$ , ..., Counter  $k$ , ..., Counter  $r$ , ...]

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Putting symbol  $Z_r$  on the top of the stack

Before: stack symbols' value =  $j$   
 After: stack symbols' value =  $j k + r$

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

[  $q$ , ..., Counter  $k$ , ..., Counter  $r$ , ...]

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----



# Counter Machine

## Stack machine



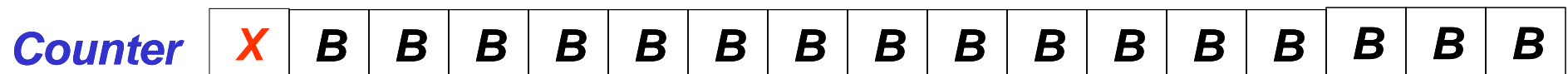
## Counter machine

Putting symbol  $Z_r$  on the top of the stack

Before: stack symbols' value =  $j$   
 After: stack symbols' value =  $j k + r$



[  $q, \dots, \text{Counter } k, \dots, \text{Counter } r, \dots$  ]



# Counter Machine

## Stack machine



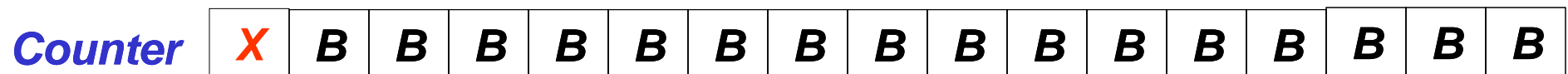
## Counter machine

Putting symbol  $Z_r$  on the top of the stack

Before: stack symbols' value =  $j$   
 After: stack symbols' value =  $j k + r$



[  $q$ , ..., Counter  $k$ , ..., Counter  $r$ , ... ]



# Counter Machine

## Stack machine



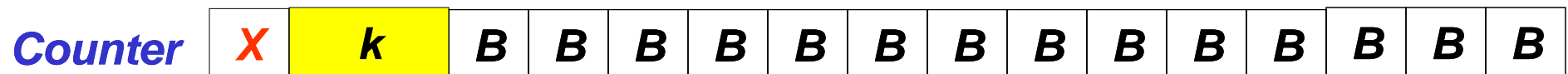
## Counter machine

Putting symbol  $Z_r$  on the top of the stack

Before: stack symbols' value =  $j$   
 After: stack symbols' value =  $j k + r$



[  $q, \dots, \text{Counter } k, \dots, \text{Counter } r, \dots$  ]



# Counter Machine

## Stack machine



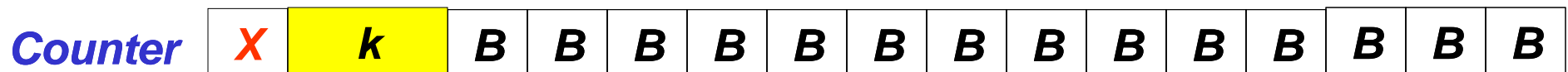
## Counter machine

Putting symbol  $Z_r$  on the top of the stack

Before: stack symbols' value =  $j$   
 After: stack symbols' value =  $j k + r$



[  $q, \dots, \text{Counter } k, \dots, \text{Counter } r, \dots$  ]



# Counter Machine

## Stack machine



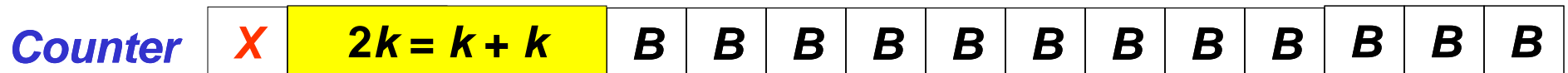
## Counter machine

Putting symbol  $Z_r$  on the top of the stack

Before: stack symbols' value =  $j$   
 After: stack symbols' value =  $j k + r$



[  $q$ , ...,  $\text{Counter } k$ , ...,  $\text{Counter } r$ , ... ]



# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Putting symbol  $Z_r$  on the top of the stack

Before: stack symbols' value =  $j$   
 After: stack symbols' value =  $j k + r$

Counter	$X$	$j - 3$					$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	---------	--	--	--	--	-----	-----	-----	-----	-----	-----	-----

[  $q$ , ...,  $\text{Counter } k$ , ...,  $\text{Counter } r$ , ... ]

Counter	$X$	$2k = k + k$			$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	--------------	--	--	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Putting symbol  $Z_r$  on the top of the stack

Before: stack symbols' value =  $j$   
 After: stack symbols' value =  $j k + r$

Counter	$X$	$j - 3$					$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	---------	--	--	--	--	-----	-----	-----	-----	-----	-----	-----

[  $q, \dots, \text{Counter } k, \dots, \text{Counter } r, \dots$  ]

Counter	$X$	$3k = 2k + k$					$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	---------------	--	--	--	--	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Putting symbol  $Z_r$  on the top of the stack

Before: stack symbols' value =  $j$   
 After: stack symbols' value =  $j k + r$

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

[  $q$ , ..., Counter  $k$ , ..., Counter  $r$ , ...]

Counter	$X$	$j k = (j-1)k + k$										$B$	$B$	$B$
---------	-----	--------------------	--	--	--	--	--	--	--	--	--	-----	-----	-----



# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Putting symbol  $Z_r$  on the top of the stack

Before: stack symbols' value =  $j$   
 After: stack symbols' value =  $j k + r$

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

[  $q, \dots, \text{Counter } k, \dots, \text{Counter } r, \dots$  ]

Counter	$X$	$j k = (j-1)k + k$										$B$	$B$	$B$
---------	-----	--------------------	--	--	--	--	--	--	--	--	--	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Putting symbol  $Z_r$  on the top of the stack

Before: stack symbols' value =  $j$   
 After: stack symbols' value =  $j k + r$

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

[  $q, \dots, \text{Counter } k, \dots, \text{Counter } r, \dots$  ]

Counter	$X$	$j k + r$														$B$
---------	-----	-----------	--	--	--	--	--	--	--	--	--	--	--	--	--	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

[  $q$ , ..., **Counter  $k$** , ..., Counter  $r$ , ...]

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Removing symbol  $Z_r$  from the top of the stack

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

[  $q, \dots, \text{Counter } k, \dots, \text{Counter } r, \dots$  ]

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Removing symbol  $Z_r$  from the top of the stack

Before: stack symbols' value =  $j$

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

[  $q, \dots, \text{Counter } k, \dots, \text{Counter } r, \dots$  ]

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Removing symbol  $Z_r$  from the top of the stack

Before: stack symbols' value =  $j$   
 After: stack symbols' value =  $\lfloor j / k \rfloor$

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

[  $q, \dots, \text{Counter } k, \dots, \text{Counter } r, \dots$  ]

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Removing symbol  $Z_r$  from the top of the stack

Before: stack symbols' value =  $j$   
 After: stack symbols' value =  $\lfloor j / k \rfloor$

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

[  $q, \dots, \text{Counter } k, \dots, \text{Counter } r, \dots$  ]

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Removing symbol  $Z_r$  from the top of the stack

Before: stack symbols' value =  $j$   
 After: stack symbols' value =  $[j / k]$

Counter	$X$	$j$					$B$	$B$	$B$	$B$	$B$
---------	-----	-----	--	--	--	--	-----	-----	-----	-----	-----

$[q, \dots, \text{Counter } k, \dots, \text{Counter } r, \dots]$

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----



# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Removing symbol  $Z_r$  from the top of the stack

Before: stack symbols' value =  $j$   
 After: stack symbols' value =  $\lfloor j/k \rfloor$

Counter	$X$	$j - k$					$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	---------	--	--	--	--	-----	-----	-----	-----	-----	-----

[  $q, \dots, \text{Counter } k, \dots, \text{Counter } r, \dots$  ]

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Removing symbol  $Z_r$  from the top of the stack

Before: stack symbols' value =  $j$   
 After: stack symbols' value =  $\lfloor j/k \rfloor$

Counter	$X$	$j - k$					$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	---------	--	--	--	--	-----	-----	-----	-----	-----	-----

[  $q, \dots, \text{Counter } k, \dots, \text{Counter } r, \dots$  ]

Counter	$X$	1	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	---	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Removing symbol  $Z_r$  from the top of the stack

Before: stack symbols' value =  $j$   
 After: stack symbols' value =  $\lfloor j/k \rfloor$

Counter	$X$	$j - 2k$					$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	----------	--	--	--	--	-----	-----	-----	-----	-----	-----	-----

[  $q, \dots, \text{Counter } k, \dots, \text{Counter } r, \dots$  ]

Counter	$X$	1	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	---	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Removing symbol  $Z_r$  from the top of the stack

Before: stack symbols' value =  $j$   
 After: stack symbols' value =  $\lfloor j/k \rfloor$

Counter	$X$	$j - 2k$					$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	----------	--	--	--	--	-----	-----	-----	-----	-----	-----	-----

[  $q, \dots, \text{Counter } k, \dots, \text{Counter } r, \dots$  ]

Counter	$X$	2	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	---	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Removing symbol  $Z_r$  from the top of the stack

Before: stack symbols' value =  $j$   
 After: stack symbols' value =  $\lfloor j/k \rfloor$

Counter	$X$	$j - 3k$					$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	----------	--	--	--	--	-----	-----	-----	-----	-----	-----	-----	-----

[  $q, \dots, \text{Counter } k, \dots, \text{Counter } r, \dots$  ]

Counter	$X$	2	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	---	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Removing symbol  $Z_r$  from the top of the stack

Before: stack symbols' value =  $j$   
 After: stack symbols' value =  $\lfloor j/k \rfloor$

Counter	$X$	$j - 3k$					$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	----------	--	--	--	--	-----	-----	-----	-----	-----	-----	-----	-----

[  $q, \dots, \text{Counter } k, \dots, \text{Counter } r, \dots$  ]

Counter	$X$	3	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	---	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Removing symbol  $Z_r$  from the top of the stack

Before: stack symbols' value =  $j$   
 After: stack symbols' value =  $\lfloor j / k \rfloor$

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

[  $q, \dots, \text{Counter } k, \dots, \text{Counter } r, \dots$  ]

Counter	$X$	$\lfloor j / k \rfloor$														$B$
---------	-----	-------------------------	--	--	--	--	--	--	--	--	--	--	--	--	--	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

[  $q$ , ..., Counter  $k$ , ..., Counter  $r$ , ..., Mod  $k$  counter ]

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----



# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Determining symbol at the top of the stack

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

[  $q$ , ..., Counter  $k$ , ..., Counter  $r$ , ..., Mod  $k$  counter ]

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Determining symbol at the top of the stack

$$j = i_m + ki_{m-1} + k^2i_{m-2} + k^3i_{m-3} + \dots + k^{m-1}i_1 = i_m + k(i_{m-1} + ki_{m-2} + \dots + k^{m-2}i_1)$$

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

[  $q$ , ..., Counter  $k$ , ..., Counter  $r$ , ..., Mod  $k$  counter ]

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Determining symbol at the top of the stack

$$j = i_m + ki_{m-1} + k^2i_{m-2} + k^3i_{m-3} + \dots + k^{m-1}i_1 = i_m + k(i_{m-1} + ki_{m-2} + \dots + k^{m-2}i_1)$$

value of the symbol at the top of the stack =  $j \bmod k$

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

[  $q$ , ..., Counter  $k$ , ..., Counter  $r$ , ..., Mod  $k$  counter ]

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Determining symbol at the top of the stack

$$j = i_m + ki_{m-1} + k^2 i_{m-2} + k^3 i_{m-3} + \dots + k^{m-1} i_1 = i_m + k(i_{m-1} + ki_{m-2} + \dots + k^{m-2} i_1)$$

value of the symbol at the top of the stack =  $j \bmod k$

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

[  $q$ , ..., Counter  $k$ , ..., Counter  $r$ , . Mod  $k$  counter  $r$  ]

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Determining symbol at the top of the stack

$$j = i_m + ki_{m-1} + k^2 i_{m-2} + k^3 i_{m-3} + \dots + k^{m-1} i_1 = i_m + k(i_{m-1} + ki_{m-2} + \dots + k^{m-2} i_1)$$

value of the symbol at the top of the stack =  $j \bmod k$

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

[  $q$ , ..., Counter  $k$ , ..., Counter  $r$ , . 0  $r$  ]

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Determining symbol at the top of the stack

$$j = i_m + ki_{m-1} + k^2i_{m-2} + k^3i_{m-3} + \dots + k^{m-1}i_1 = i_m + k(i_{m-1} + ki_{m-2} + \dots + k^{m-2}i_1)$$

value of the symbol at the top of the stack =  $j \bmod k$

Counter	$X$	$j$					$B$	$B$	$B$	$B$	$B$
---------	-----	-----	--	--	--	--	-----	-----	-----	-----	-----

[  $q$ , ..., Counter  $k$ , ..., Counter  $r$ , . 0  $r$  ]

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Determining symbol at the top of the stack

$$j = i_m + ki_{m-1} + k^2 i_{m-2} + k^3 i_{m-3} + \dots + k^{m-1} i_1 = i_m + k(i_{m-1} + ki_{m-2} + \dots + k^{m-2} i_1)$$

value of the symbol at the top of the stack =  $j \bmod k$

Counter	$X$	$j - 1$					$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	---------	--	--	--	--	-----	-----	-----	-----	-----	-----

[  $q, \dots, \text{Counter } k, \dots, \text{Counter } r, .$   $0$   $r$  ]

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine



## Counter machine

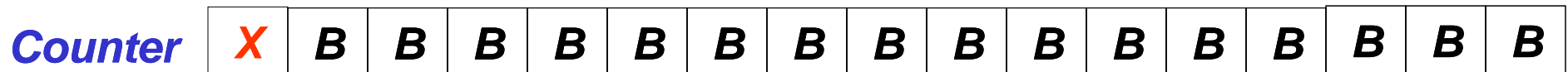
Determining symbol at the top of the stack

$$j = i_m + ki_{m-1} + k^2i_{m-2} + k^3i_{m-3} + \dots + k^{m-1}i_1 = i_m + k(i_{m-1} + ki_{m-2} + \dots + k^{m-2}i_1)$$

value of the symbol at the top of the stack =  $j \bmod k$



[  $q$ , ..., Counter  $k$ , ..., Counter  $r$ , .  $1$   $r$  ]





# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Determining symbol at the top of the stack

$$j = i_m + ki_{m-1} + k^2i_{m-2} + k^3i_{m-3} + \dots + k^{m-1}i_1 = i_m + k(i_{m-1} + ki_{m-2} + \dots + k^{m-2}i_1)$$

value of the symbol at the top of the stack =  $j \bmod k$

Counter	$X$	$j - 1$					$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	---------	--	--	--	--	-----	-----	-----	-----	-----	-----

[  $q, \dots, \text{Counter } k, \dots, \text{Counter } r, .$   $1$   $r$  ]

Counter	$X$	$1$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Determining symbol at the top of the stack

$$j = i_m + ki_{m-1} + k^2 i_{m-2} + k^3 i_{m-3} + \dots + k^{m-1} i_1 = i_m + k(i_{m-1} + ki_{m-2} + \dots + k^{m-2} i_1)$$

value of the symbol at the top of the stack =  $j \bmod k$

Counter	$X$	$j - 2$					$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	---------	--	--	--	--	-----	-----	-----	-----	-----	-----	-----

[  $q, \dots, \text{Counter } k, \dots, \text{Counter } r, .$   $1$   $r$  ]

Counter	$X$	$1$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Determining symbol at the top of the stack

$$j = i_m + ki_{m-1} + k^2 i_{m-2} + k^3 i_{m-3} + \dots + k^{m-1} i_1 = i_m + k(i_{m-1} + ki_{m-2} + \dots + k^{m-2} i_1)$$

value of the symbol at the top of the stack =  $j \bmod k$

Counter	$X$	$j - 2$					$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	---------	--	--	--	--	-----	-----	-----	-----	-----	-----	-----

[  $q, \dots, \text{Counter } k, \dots, \text{Counter } r, .$   $2$   $r$  ]

Counter	$X$	1	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	---	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Determining symbol at the top of the stack

$$j = i_m + ki_{m-1} + k^2 i_{m-2} + k^3 i_{m-3} + \dots + k^{m-1} i_1 = i_m + k(i_{m-1} + ki_{m-2} + \dots + k^{m-2} i_1)$$

value of the symbol at the top of the stack =  $j \bmod k$

Counter	$X$	$j - 2$					$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	---------	--	--	--	--	-----	-----	-----	-----	-----	-----	-----

[  $q, \dots, \text{Counter } k, \dots, \text{Counter } r, .$   $2$   $r$  ]

Counter	$X$	$2$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Determining symbol at the top of the stack

$$j = i_m + ki_{m-1} + k^2i_{m-2} + k^3i_{m-3} + \dots + k^{m-1}i_1 = i_m + k(i_{m-1} + ki_{m-2} + \dots + k^{m-2}i_1)$$

value of the symbol at the top of the stack =  $j \bmod k$

Counter	$X$	$j - 3$					$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	---------	--	--	--	--	-----	-----	-----	-----	-----	-----	-----

[  $q$ , ..., Counter  $k$ , ..., Counter  $r$ , .  $2$   $r$  ]

Counter	$X$	$2$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Determining symbol at the top of the stack

$$j = i_m + ki_{m-1} + k^2 i_{m-2} + k^3 i_{m-3} + \dots + k^{m-1} i_1 = i_m + k(i_{m-1} + ki_{m-2} + \dots + k^{m-2} i_1)$$

value of the symbol at the top of the stack =  $j \bmod k$

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

[  $q$ , ..., Counter  $k$ , ..., Counter  $r$ , . 3  $r$  ]

Counter	$X$	2	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	---	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Determining symbol at the top of the stack

$$j = i_m + ki_{m-1} + k^2 i_{m-2} + k^3 i_{m-3} + \dots + k^{m-1} i_1 = i_m + k(i_{m-1} + ki_{m-2} + \dots + k^{m-2} i_1)$$

value of the symbol at the top of the stack =  $j \bmod k$

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

[  $q$ , ..., Counter  $k$ , ..., Counter  $r$ , . 3  $r$  ]

Counter	$X$	3	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	---	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Determining symbol at the top of the stack

$$j = i_m + ki_{m-1} + k^2 i_{m-2} + k^3 i_{m-3} + \dots + k^{m-1} i_1 = i_m + k(i_{m-1} + ki_{m-2} + \dots + k^{m-2} i_1)$$

value of the symbol at the top of the stack =  $j \bmod k$

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

[  $q$ , ..., Counter  $k$ , ..., Counter  $r$ , .  $k-3$  ,  $r$  ]

Counter	$X$	3	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	---	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----



# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Determining symbol at the top of the stack

$$j = i_m + ki_{m-1} + k^2 i_{m-2} + k^3 i_{m-3} + \dots + k^{m-1} i_1 = i_m + k(i_{m-1} + ki_{m-2} + \dots + k^{m-2} i_1)$$

value of the symbol at the top of the stack =  $j \bmod k$

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

[  $q$ , ..., Counter  $k$ , ..., Counter  $r$ , .  $k-2$  ,  $r$  ]

Counter	$X$	3	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	---	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Determining symbol at the top of the stack

$$j = i_m + ki_{m-1} + k^2 i_{m-2} + k^3 i_{m-3} + \dots + k^{m-1} i_1 = i_m + k(i_{m-1} + ki_{m-2} + \dots + k^{m-2} i_1)$$

value of the symbol at the top of the stack =  $j \bmod k$

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

[  $q$ , ..., Counter  $k$ , ..., Counter  $r$ , .  $k-1$  ,  $r$  ]

Counter	$X$	3	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	---	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Determining symbol at the top of the stack

$$j = i_m + ki_{m-1} + k^2 i_{m-2} + k^3 i_{m-3} + \dots + k^{m-1} i_1 = i_m + k(i_{m-1} + ki_{m-2} + \dots + k^{m-2} i_1)$$

value of the symbol at the top of the stack =  $j \bmod k$

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

[  $q$ , ..., Counter  $k$ , ..., Counter  $r$ , . 0  $r$  ]

Counter	$X$	3	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	---	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Determining symbol at the top of the stack

$$j = i_m + ki_{m-1} + k^2i_{m-2} + k^3i_{m-3} + \dots + k^{m-1}i_1 = i_m + k(i_{m-1} + ki_{m-2} + \dots + k^{m-2}i_1)$$

value of the symbol at the top of the stack =  $j \bmod k$

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

[  $q$ , ..., Counter  $k$ , ..., Counter  $r$ , .  $1$   $r$  ]

Counter	$X$	$3$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Determining symbol at the top of the stack

$$j = i_m + ki_{m-1} + k^2i_{m-2} + k^3i_{m-3} + \dots + k^{m-1}i_1 = i_m + k(i_{m-1} + ki_{m-2} + \dots + k^{m-2}i_1)$$

value of the symbol at the top of the stack =  $j \bmod k$

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

[  $q$ , ..., Counter  $k$ , ..., Counter  $r$ , .  $2$   $r$  ]

Counter	$X$	$3$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Counter Machine

## Stack machine

Stack	$Z_{i1}$	$Z_{i2}$	$Z_{i3}$	$Z_{i4}$	- -	$Z_{im}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	----------	----------	----------	----------	-----	----------	-----	-----	-----	-----	-----	-----	-----	-----

## Counter machine

Determining symbol at the top of the stack

$$j = i_m + ki_{m-1} + k^2 i_{m-2} + k^3 i_{m-3} + \dots + k^{m-1} i_1 = i_m + k(i_{m-1} + ki_{m-2} + \dots + k^{m-2} i_1)$$

value of the symbol at the top of the stack =  $j \bmod k$

Counter	$X$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

[  $q$ , ..., Counter  $k$ , ..., Counter  $r$ , . 2  $r$  ]

Counter	$X$	$j$													$B$
---------	-----	-----	--	--	--	--	--	--	--	--	--	--	--	--	-----

# Counter Machine

# Counter Machine

*Machine with 4 counters*



# Counter Machine

## Machine with 4 counters

Counter	<b>X</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
Counter	<b>X</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
Counter	<b>X</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
Counter	<b>X</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>

---

# Counter Machine

## Machine with 4 counters

Counter	X	B	B	B	B	B	B	B	B	B	B	B	B
Counter	X	B	B	B	B	B	B	B	B	B	B	B	B
Counter	X	B	B	B	B	B	B	B	B	B	B	B	B
Counter	X	B	B	B	B	B	B	B	B	B	B	B	B

## Machine with 2 counters

# Counter Machine

## Machine with 4 counters

Counter	X	B	B	B	B	B	B	B	B	B	B	B	B
Counter	X	B	B	B	B	B	B	B	B	B	B	B	B
Counter	X	B	B	B	B	B	B	B	B	B	B	B	B
Counter	X	B	B	B	B	B	B	B	B	B	B	B	B

## Machine with 2 counters

Counter	X	B	B	B	B	B	B	B	B	B	B	B	B
Counter	X	B	B	B	B	B	B	B	B	B	B	B	B

# Counter Machine

## Machine with 4 counters

Counter	X	i				B	B	B	B	B	B	B	B
Counter	X	B	B	B	B	B	B	B	B	B	B	B	B
Counter	X	B	B	B	B	B	B	B	B	B	B	B	B
Counter	X	B	B	B	B	B	B	B	B	B	B	B	B

## Machine with 2 counters

Counter	X	B	B	B	B	B	B	B	B	B	B	B	B
Counter	X	B	B	B	B	B	B	B	B	B	B	B	B

# Counter Machine

## Machine with 4 counters

Counter	X	i				B	B	B	B	B	B	B	B
Counter	X	j						B	B	B	B	B	B
Counter	X	B	B	B	B	B	B	B	B	B	B	B	B
Counter	X	B	B	B	B	B	B	B	B	B	B	B	B

## Machine with 2 counters

Counter	X	B	B	B	B	B	B	B	B	B	B	B	B
Counter	X	B	B	B	B	B	B	B	B	B	B	B	B

# Counter Machine

## Machine with 4 counters

Counter	<b>X</b>	<b><i>i</i></b>				<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
Counter	<b>X</b>	<b><i>j</i></b>						<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
Counter	<b>X</b>	<b><i>k</i></b>		<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
Counter	<b>X</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>

## Machine with 2 counters

Counter	<b>X</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
Counter	<b>X</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>

# Counter Machine

## Machine with 4 counters

Counter	<b>X</b>	<b><i>i</i></b>				<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
Counter	<b>X</b>	<b><i>j</i></b>						<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
Counter	<b>X</b>	<b><i>k</i></b>		<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
Counter	<b>X</b>	<b><i>l</i></b>								<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>

## Machine with 2 counters

Counter	<b>X</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
Counter	<b>X</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>

# Counter Machine

## Machine with 4 counters

Counter	<b>X</b>	<b><i>i</i></b>				<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
Counter	<b>X</b>	<b><i>j</i></b>						<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
Counter	<b>X</b>	<b><i>k</i></b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
Counter	<b>X</b>	<b><i>l</i></b>							<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>

## Machine with 2 counters

Counter	<b>X</b>	<b><math>n = 2^i 3^j 5^k 7^l</math></b>										<b>B</b>
Counter	<b>X</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>



# Counter Machine

## Machine with 4 counters

Counter	X	<i>i</i>				B	B	B	B	B	B	B	B
Counter	X	<i>j</i>						B	B	B	B	B	B
Counter	X	<i>k</i>	B	B	B	B	B	B	B	B	B	B	B
Counter	X	<i>l</i>							B	B	B	B	

## Machine with 2 counters

Counter	X	$n = 2^i 3^j 5^k 7^l$										B
Counter	X	B	B	B	B	B	B	B	B	B	B	B

$$1) (i, j, k, l) = \pm 1$$

# Counter Machine

## Machine with 4 counters

Counter	X	i				B	B	B	B	B	B	B	B
Counter	X	j						B	B	B	B	B	B
Counter	X	k	B	B	B	B	B	B	B	B	B	B	B
Counter	X	l							B	B	B	B	

## Machine with 2 counters

Counter	X	$n = 2^i 3^j 5^k 7^l$										B
Counter	X	B	B	B	B	B	B	B	B	B	B	B

$$1) (i, j, k, l) = \pm 1$$

- multiply or divide value  $n$  by an appropriate value 2, 3, 5 or 7

# Counter Machine

## Machine with 4 counters

Counter	X	$i$				B	B	B	B	B	B	B	B
Counter	X	$j$						B	B	B	B	B	B
Counter	X	$k$	B	B	B	B	B	B	B	B	B	B	B
Counter	X	$l$							B	B	B	B	

## Machine with 2 counters

Counter	X	$n = 2^i 3^j 5^k 7^l$										B	
Counter	X	B	B	B	B	B	B	B	B	B	B	B	B

# Counter Machine

## Machine with 4 counters

Counter	X	$i$				B	B	B	B	B	B	B	B
Counter	X	$j$						B	B	B	B	B	B
Counter	X	$k$	B	B	B	B	B	B	B	B	B	B	B
Counter	X	$l$						B	B	B	B		

## Machine with 2 counters

Counter	X	$n = 2^i 3^j 5^k 7^l$										B
Counter	X	B	B	B	B	B	B	B	B	B	B	B

2)  $(i, j, k, l) == 0$  ?

# Counter Machine

## Machine with 4 counters

Counter	X	i				B	B	B	B	B	B	B	B
Counter	X	j						B	B	B	B	B	B
Counter	X	k	B	B	B	B	B	B	B	B	B	B	B
Counter	X	l						B	B	B	B		

## Machine with 2 counters

Counter	X	$n = 2^i 3^j 5^k 7^l$										B
Counter	X	B	B	B	B	B	B	B	B	B	B	B

2) ( i, j, k, l ) == 0 ?

- divide  $n$  by 2, 3, 5 or 7

# Counter Machine

## Machine with 4 counters

Counter	X	<i>i</i>				B	B	B	B	B	B	B	B
Counter	X	<i>j</i>						B	B	B	B	B	B
Counter	X	<i>k</i>	B	B	B	B	B	B	B	B	B	B	B
Counter	X	<i>l</i>							B	B	B	B	

## Machine with 2 counters

Counter	X	$n = 2^i 3^j 5^k 7^l$										B
Counter	X	B	B	B	B	B	B	B	B	B	B	B

2) ( *i*, *j*, *k*, *l* ) == 0 ?

- divide *n* by 2, 3, 5 or 7

- (division remainder == 0) ?

# Counter Machine

## Machine with 4 counters

Counter	X	i				B	B	B	B	B	B	B	B
Counter	X	j						B	B	B	B	B	B
Counter	X	k	B	B	B	B	B	B	B	B	B	B	B
Counter	X	l							B	B	B	B	

## Machine with 2 counters

Counter	X	$n = 2^i 3^j 5^k 7^l$										B
Counter	X	B	B	B	B	B	B	B	B	B	B	B

2)  $(i, j, k, l) == 0$  ?

- divide  $n$  by 2, 3, 5 or 7

- (division remainder == 0) ?

$(i, j, k, l) \neq 0$

# TM with a limited number of states and tape symbols



# TM with a limited number of states and tape symbols

- **Acceptance of any language**

# TM with a limited number of states and tape symbols

- **Acceptance of any language**
- **Unlimited number of tape symbols**

# TM with a limited number of states and tape symbols

- **Acceptance of any language**
- **Unlimited number of tape symbols**
  - Single tape

# TM with a limited number of states and tape symbols

- **Acceptance of any language**
- **Unlimited number of tape symbols**
  - Single tape
  - Three states

# TM with a limited number of states and tape symbols

- **Acceptance of any language**
- **Unlimited number of tape symbols**
  - Single tape
  - Three states
    - Two non accepting states

# TM with a limited number of states and tape symbols

- **Acceptance of any language**
- **Unlimited number of tape symbols**
  - Single tape
  - Three states
    - Two non accepting states
    - One accepting state

# TM with a limited number of states and tape symbols

# TM with a limited number of states and tape symbols

- **Unlimited number of states**



# TM with a limited number of states and tape symbols

- **Unlimited number of states**
  - tape symbols  $\{0, 1, B\}$

# TM with a limited number of states and tape symbols

- **Unlimited number of states**

- tape symbols  $\{0, 1, B\}$
- $2^{k-1}+1 < \textit{cardinal number of tape symbol set} < 2^k$

# TM with a limited number of states and tape symbols

- **Unlimited number of states**
  - tape symbols  $\{0, 1, B\}$
  - $2^{k-1}+1 < \text{cardinal number of tape symbol set} < 2^k$
  - tape symbols are coded by binary strings of length  $k$

# TM with a limited number of states and tape symbols

- **Unlimited number of states**

- tape symbols  $\{0, 1, B\}$
- $2^{k-1}+1 < \text{cardinal number of tape symbol set} < 2^k$
- tape symbols are coded by binary strings of length  $k$
- $k$  tape cells contain  $k$ -bit binary code

# TM with a limited number of states and tape symbols

- **Unlimited number of states**

- tape symbols  $\{0, 1, B\}$
- $2^{k-1}+1 < \text{cardinal number of tape symbol set} < 2^k$
- tape symbols are coded by binary strings of length  $k$
- $k$  tape cells contain  $k$ -bit binary code
- one state component - state TM  $M_1$

# TM with a limited number of states and tape symbols

- **Unlimited number of states**

- tape symbols  $\{0, 1, B\}$
- $2^{k-1}+1 < \text{cardinal number of tape symbol set} < 2^k$
- tape symbols are coded by binary strings of length  $k$
- $k$  tape cells contain  $k$ -bit binary code
- one state component - state TM  $M_1$
- another state component – sets head at the beginning of the binary code

# TM with a limited number of states and tape symbols

- **Unlimited number of states**

- tape symbols  $\{0, 1, B\}$
- $2^{k-1}+1 < \text{cardinal number of tape symbol set} < 2^k$
- tape symbols are coded by binary strings of length  $k$
- $k$  tape cells contain  $k$ -bit binary code
- one state component - state TM  $M_1$
- another state component – sets head at the beginning of the binary code
- empty cell symbol  $B$

# TM with a limited number of states and tape symbols

- **Unlimited number of states**

- tape symbols  $\{0, 1, B\}$
- $2^{k-1}+1 < \text{cardinal number of tape symbol set} < 2^k$
- tape symbols are coded by binary strings of length  $k$
- $k$  tape cells contain  $k$ -bit binary code
- one state component - state TM  $M_1$
- another state component – sets head at the beginning of the binary code
- empty cell symbol  $B$ 
  - TM  $M_2$  changes  $k$  empty cell symbols  $B$  with binary code for an empty cell



# TM with a limited number of states and tape symbols

- **Unlimited number of states**

- tape symbols  $\{0, 1, B\}$
- $2^{k-1}+1 < \text{cardinal number of tape symbol set} < 2^k$
- tape symbols are coded by binary strings of length  $k$
- $k$  tape cells contain  $k$ -bit binary code
- one state component - state TM  $M_1$
- another state component – sets head at the beginning of the binary code
- empty cell symbol  $B$ 
  - TM  $M_2$  changes  $k$  empty cell symbols  $B$  with binary code for an empty cell
  - TM  $M_1$  simulation continues

# Universal Turing Machine $M_u$

# Universal Turing Machine $M_u$

*Tape 1*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

*Tape 2*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

*Tape 3*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

# Universal Turing Machine $M_u$

***Tape 1***

***Transition function TM M***

***B B B B B***

***Tape 2***

***B B B B B B B B B B B B B***

***Tape 3***

***B B B B B B B B B B B B B***

# Universal Turing Machine $M_u$

***Tape 1***

<b><i>Transition function TM M</i></b>	<b><i>w</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
--	-----------------	-----------------	-----------------

***Tape 2***

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

***Tape 3***

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

# Universal Turing Machine $M_u$

**Tape 1**

<b>Transition function TM <math>M</math></b>	<b>w</b>	<b>B</b>	<b>B</b>
--	----------	----------	----------

**Tape 2**

<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

**Tape 3**

<b>State TM <math>M</math></b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
--------------------------------	----------	----------	----------	----------	----------	----------	----------	----------

# Universal Turing Machine $M_u$

**Tape 1**

<b>Transition function TM <math>M</math></b>	<b>w</b>	<b>B</b>	<b>B</b>
--	----------	----------	----------

**Tape 2**

<b>Tape content TM <math>M</math></b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
---------------------------------------	----------	----------	----------	----------	----------	----------	----------

**Tape 3**

<b>State TM <math>M</math></b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
--------------------------------	----------	----------	----------	----------	----------	----------	----------

# Universal Turing Machine $M_u$

***Tape 1***

<b><i>Transition function TM M</i></b>	<b><i>w</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
--	-----------------	-----------------	-----------------

***Tape 2***

<b><i>w</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

***Tape 3***

<b><i>State TM M</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
--------------------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------



# Universal Turing Machine $M_u$

***Tape 1***

<b><i>Transition function TM M</i></b>	<b><i>w</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
--	-----------------	-----------------	-----------------

***Tape 2***

<b><i>w</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

***Tape 3***

<b><i>q<sub>0</sub></i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

# Universal Turing Machine $M_u$

***Tape 1***

<b><i>Transition function TM <math>M</math></i></b>	<b><i>w</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
---	-----------------	-----------------	-----------------

***Tape 2***

<b><i>t</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------

***Tape 3***

<b><i>q<sub>0</sub></i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

# Universal Turing Machine $M_u$

***Tape 1***

<b><i>Transition function TM M</i></b>	<b><i>w</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
--	-----------------	-----------------	-----------------

***Tape 2***

<b><i>t</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------

***Tape 3***

<b><i>q</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

# Universal Turing Machine $M_u$

**Tape 1**

<b>Transition function TM <math>M</math></b>	<b>w</b>	<b>B</b>	<b>B</b>
--	----------	----------	----------

**Tape 2**

<b>t</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------	----------	----------

**Tape 3**

<b>q</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------	----------	----------	----------	----------	----------	----------

- 
- It is possible to construct universal TM  $M_u$  using:

# Universal Turing Machine $M_u$

<b>Tape 1</b>	<b>Transition function TM <math>M</math></b>	<b>w</b>	<b>B</b>	<b>B</b>				
<b>Tape 2</b>	<b>t</b>		<b>B</b>	<b>B</b>	<b>B</b>			
<b>Tape 3</b>	<b>q</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>

- It is possible to construct universal TM  $M_u$  using:
  - Single tape

# Universal Turing Machine $M_u$

<b>Tape 1</b>	<b>Transition function TM <math>M</math></b>	<b>w</b>	<b>B</b>	<b>B</b>				
<b>Tape 2</b>	<b>t</b>			<b>B</b>	<b>B</b>	<b>B</b>		
<b>Tape 3</b>	<b>q</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>

- It is possible to construct universal TM  $M_u$  using:
  - Single tape
  - Five states

# Universal Turing Machine $M_u$

<b>Tape 1</b>	<b>Transition function <math>TM\ M</math></b>	<b>w</b>	<b>B</b>	<b>B</b>				
<b>Tape 2</b>	<b>t</b>		<b>B</b>	<b>B</b>	<b>B</b>			
<b>Tape 3</b>	<b>q</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>

- It is possible to construct universal TM  $M_u$  using:
  - Single tape
  - Five states
  - Five tape symbols

# Lecture 13

## 4.1.4 Simplified Turing Machine Models

## 4.1.5 Generating Languages Using Turing Machines



# Generating Languages Using Turing Machines

# Generating Languages Using Turing Machines

*Storage tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

*Storage tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

*Output tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

# Generating Languages Using Turing Machines

*Storage tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

*Storage tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

*Output tape*

<b>#</b>	<b><math>w_1</math></b>	<b>#</b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
----------	-------------------------	----------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

# Generating Languages Using Turing Machines

*Storage tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

*Storage tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

*Output tape*

<b>#</b>	<b><math>w_1</math></b>	<b>#</b>	<b><math>w_2</math></b>	<b>#</b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
----------	-------------------------	----------	-------------------------	----------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

# Generating Languages Using Turing Machines

*Storage tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

*Storage tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

*Output tape*

<b><i>#</i></b>	<b><i>w<sub>1</sub></i></b>	<b><i>#</i></b>	<b><i>w<sub>2</sub></i></b>	<b><i>#</i></b>	<b><i>w<sub>3</sub></i></b>	<b><i>#</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------------------	-----------------	-----------------------------	-----------------	-----------------------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

# Generating Languages Using Turing Machines

*Storage tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

*Storage tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

*Output tape*

<b><i>#</i></b>	<b><i>w<sub>1</sub></i></b>	<b><i>#</i></b>	<b><i>w<sub>2</sub></i></b>	<b><i>#</i></b>	<b><i>w<sub>3</sub></i></b>	<b><i>#</i></b>	<b><i>w<sub>4</sub></i></b>	<b><i>#</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------------------	-----------------	-----------------------------	-----------------	-----------------------------	-----------------	-----------------------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

# Generating Languages Using Turing Machines

*Storage tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

*Storage tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

*Output tape*

<b><i>#</i></b>	<b><i>w<sub>1</sub></i></b>	<b><i>#</i></b>	<b><i>w<sub>2</sub></i></b>	<b><i>#</i></b>	<b><i>w<sub>3</sub></i></b>	<b><i>#</i></b>	<b><i>w<sub>4</sub></i></b>	<b><i>#</i></b>	<b><i>w<sub>5</sub></i></b>	<b><i>#</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------------------	-----------------	-----------------------------	-----------------	-----------------------------	-----------------	-----------------------------	-----------------	-----------------------------	-----------------	-----------------	-----------------	-----------------

# Generating Languages Using Turing Machines

*Storage tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

*Storage tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

*Output tape*

<b>#</b>	<b><math>w_1</math></b>	<b>#</b>	<b><math>w_2</math></b>	<b>#</b>	<b><math>w_3</math></b>	<b>#</b>	<b><math>w_4</math></b>	<b>#</b>	<b><math>w_5</math></b>	<b>#</b>	<b><math>w_6</math></b>	<b>#</b>	<b><i>B</i></b>
----------	-------------------------	----------	-------------------------	----------	-------------------------	----------	-------------------------	----------	-------------------------	----------	-------------------------	----------	-----------------



# Generating Languages Using Turing Machines

*Storage tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

*Storage tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

*Output tape*

<b><i>#</i></b>	<b><i>w<sub>1</sub></i></b>	<b><i>#</i></b>	<b><i>w<sub>2</sub></i></b>	<b><i>#</i></b>	<b><i>w<sub>3</sub></i></b>	<b><i>#</i></b>	<b><i>w<sub>4</sub></i></b>	<b><i>#</i></b>	<b><i>w<sub>5</sub></i></b>	<b><i>#</i></b>	<b><i>w<sub>6</sub></i></b>	<b><i>#</i></b>	<b><i>w<sub>7</sub></i></b>	<b><i>#</i></b>
-----------------	-----------------------------	-----------------	-----------------------------	-----------------	-----------------------------	-----------------	-----------------------------	-----------------	-----------------------------	-----------------	-----------------------------	-----------------	-----------------------------	-----------------

# Accepting Language Generated by a Turing Machine

# Accepting Language Generated by a Turing Machine

*Input tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

*Storage tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

*Output tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

# Accepting Language Generated by a Turing Machine

*Input tape*

<b>w</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Storage tape*

<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Output tape*

<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

# Accepting Language Generated by a Turing Machine

*Input tape*

<b><i>w</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

*Storage tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

*Output tape*

<b>#</b>	<b><i>w</i><sub>1</sub></b>	<b>#</b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
----------	-----------------------------	----------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

# Accepting Language Generated by a Turing Machine

*Input tape*

<b>w</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Storage tape*

<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Output tape*

<b>#w</b>	<b>w<sub>1</sub></b>	<b>#</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
-----------	----------------------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

# Accepting Language Generated by a Turing Machine

*Input tape*

<b>w</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Storage tape*

<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Output tape*

<b>#</b>	<b>w<sub>1</sub></b>	<b>#</b>	<b>w<sub>2</sub></b>	<b>#</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------------------	----------	----------------------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

# Accepting Language Generated by a Turing Machine

*Input tape*

<b><i>w</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

*Storage tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

*Output tape*

<b>#</b>	<b><i>w</i><sub>1</sub></b>	<b>#</b>	<b><i>w</i><sub>2</sub></b>	<b><i>w</i></b>	<b>#</b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
----------	-----------------------------	----------	-----------------------------	-----------------	----------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------



# Accepting Language Generated by a Turing Machine

*Input tape*

<b>w</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Storage tape*

<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Output tape*

<b>#</b>	<b>w<sub>1</sub></b>	<b>#</b>	<b>w<sub>2</sub></b>	<b>#</b>	<b>w<sub>3</sub></b>	<b>#</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------------------	----------	----------------------	----------	----------------------	----------	----------	----------	----------	----------	----------	----------	----------	----------

# Accepting Language Generated by a Turing Machine

*Input tape*

<b>w</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Storage tape*

<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Output tape*

<b>#</b>	<b>w<sub>1</sub></b>	<b>#</b>	<b>w<sub>2</sub></b>	<b>#</b>	<b>w<sub>3</sub></b>	<b>w</b>	<b>#</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------------------	----------	----------------------	----------	----------------------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

# Accepting Language Generated by a Turing Machine

*Input tape*

<b>w</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Storage tape*

<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Output tape*

<b>#</b>	<b>w<sub>1</sub></b>	<b>#</b>	<b>w<sub>2</sub></b>	<b>#</b>	<b>w<sub>3</sub></b>	<b>#</b>	<b>w<sub>4</sub></b>	<b>#</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------------------	----------	----------------------	----------	----------------------	----------	----------------------	----------	----------	----------	----------	----------	----------	----------

# Accepting Language Generated by a Turing Machine

*Input tape*

<b>w</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Storage tape*

<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Output tape*

<b>#</b>	<b>w<sub>1</sub></b>	<b>#</b>	<b>w<sub>2</sub></b>	<b>#</b>	<b>w<sub>3</sub></b>	<b>#</b>	<b>w<sub>4</sub> w</b>	<b>#</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------------------	----------	----------------------	----------	----------------------	----------	------------------------	----------	----------	----------	----------	----------	----------	----------

# Accepting Language Generated by a Turing Machine

*Input tape*

<b>w</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Storage tape*

<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Output tape*

<b>#</b>	<b>w<sub>1</sub></b>	<b>#</b>	<b>w<sub>2</sub></b>	<b>#</b>	<b>w<sub>3</sub></b>	<b>#</b>	<b>w<sub>4</sub></b>	<b>#</b>	<b>w<sub>5</sub></b>	<b>#</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------------------	----------	----------------------	----------	----------------------	----------	----------------------	----------	----------------------	----------	----------	----------	----------

# Accepting Language Generated by a Turing Machine

*Input tape*

<b>w</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Storage tape*

<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Output tape*

<b>#</b>	<b>w<sub>1</sub></b>	<b>#</b>	<b>w<sub>2</sub></b>	<b>#</b>	<b>w<sub>3</sub></b>	<b>#</b>	<b>w<sub>4</sub></b>	<b>#</b>	<b>w<sub>5</sub> w</b>	<b>#</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------------------	----------	----------------------	----------	----------------------	----------	----------------------	----------	------------------------	----------	----------	----------	----------

# Accepting Language Generated by a Turing Machine

*Input tape*

<b>w</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Storage tape*

<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Output tape*

<b>#</b>	<b>w<sub>1</sub></b>	<b>#</b>	<b>w<sub>2</sub></b>	<b>#</b>	<b>w<sub>3</sub></b>	<b>#</b>	<b>w<sub>4</sub></b>	<b>#</b>	<b>w<sub>5</sub></b>	<b>#</b>	<b>w<sub>6</sub></b>	<b>#</b>	<b>B</b>
----------	----------------------	----------	----------------------	----------	----------------------	----------	----------------------	----------	----------------------	----------	----------------------	----------	----------

# Accepting Language Generated by a Turing Machine

*Input tape*

<b>w</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Storage tape*

<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Output tape*

<b>#</b>	<b>w<sub>1</sub></b>	<b>#</b>	<b>w<sub>2</sub></b>	<b>#</b>	<b>w<sub>3</sub></b>	<b>#</b>	<b>w<sub>4</sub></b>	<b>#</b>	<b>w<sub>5</sub></b>	<b>#</b>	<b>w<sub>6</sub> w</b>	<b>#</b>	<b>B</b>
----------	----------------------	----------	----------------------	----------	----------------------	----------	----------------------	----------	----------------------	----------	------------------------	----------	----------



# Accepting Language Generated by a Turing Machine

*Input tape*

<b>w</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Storage tape*

<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Output tape*

<b>#</b>	<b>w<sub>1</sub></b>	<b>#</b>	<b>w<sub>2</sub></b>	<b>#</b>	<b>w<sub>3</sub></b>	<b>#</b>	<b>w<sub>4</sub></b>	<b>#</b>	<b>w<sub>5</sub></b>	<b>#</b>	<b>w<sub>6</sub></b>	<b>#</b>	<b>w<sub>7</sub></b>	<b>#</b>
----------	----------------------	----------	----------------------	----------	----------------------	----------	----------------------	----------	----------------------	----------	----------------------	----------	----------------------	----------

# Accepting Language Generated by a Turing Machine

*Input tape*

<b>w</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Storage tape*

<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Output tape*

<b>#</b>	<b>w<sub>1</sub></b>	<b>#</b>	<b>w<sub>2</sub></b>	<b>#</b>	<b>w<sub>3</sub></b>	<b>#</b>	<b>w<sub>4</sub></b>	<b>#</b>	<b>w<sub>5</sub></b>	<b>#</b>	<b>w<sub>6</sub></b>	<b>#</b>	<b>w<sub>7</sub></b>	<b>w</b>	<b>#</b>
----------	----------------------	----------	----------------------	----------	----------------------	----------	----------------------	----------	----------------------	----------	----------------------	----------	----------------------	----------	----------

# Accepting Language Generated by a Turing Machine

*Input tape*

<b>w</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Storage tape*

<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>B</b>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Output tape*

<b>#</b>	<b>w<sub>1</sub></b>	<b>#</b>	<b>w<sub>2</sub></b>	<b>#</b>	<b>w<sub>3</sub></b>	<b>#</b>	<b>w<sub>4</sub></b>	<b>#</b>	<b>w<sub>5</sub></b>	<b>#</b>	<b>w<sub>6</sub></b>	<b>#</b>	<b>w<sub>7</sub></b>	<b>#</b>
----------	----------------------	----------	----------------------	----------	----------------------	----------	----------------------	----------	----------------------	----------	----------------------	----------	----------------------	----------

# Generating Languages Accepted by a Turing Machine

# Generating Languages Accepted by a Turing Machine

*Storage tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

*Storage tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

*Output tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

# Generating Languages Accepted by a Turing Machine

*Storage tape*

$w_1$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Storage tape*

$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Output tape*

$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Generating Languages Accepted by a Turing Machine

*Storage tape*

$w_1$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Storage tape*

<i>Simulation of TM <math>M_2</math> for string <math>w_1</math></i>												$B$	$B$
--	--	--	--	--	--	--	--	--	--	--	--	-----	-----

*Output tape*

$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Generating Languages Accepted by a Turing Machine

*Storage tape*

$w_1$	$w_2$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Storage tape*

<i>Simulation of TM <math>M_2</math> for string <math>w_1</math></i>											$B$	$B$
--	--	--	--	--	--	--	--	--	--	--	-----	-----

*Output tape*

$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----



# Generating Languages Accepted by a Turing Machine

*Storage tape*

$w_1$	$w_2$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Storage tape*

<i>Simulation of TM <math>M_2</math> for string <math>w_2</math></i>											$B$	$B$
--	--	--	--	--	--	--	--	--	--	--	-----	-----

*Output tape*

$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Generating Languages Accepted by a Turing Machine

*Storage tape*

$w_1$	$w_2$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Storage tape*

<i>Simulation of TM <math>M_2</math> for string <math>w_2</math></i>										$B$	$B$
--	--	--	--	--	--	--	--	--	--	-----	-----

*Output tape*

#	$w_2$	#	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---	-------	---	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Generating Languages Accepted by a Turing Machine

*Storage tape*

$w_1$	$w_2$	$w_3$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-------	-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Storage tape*

<i>Simulation of TM <math>M_2</math> for string <math>w_2</math></i>										$B$	$B$
--	--	--	--	--	--	--	--	--	--	-----	-----

*Output tape*

#	$w_2$	#	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---	-------	---	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Generating Languages Accepted by a Turing Machine

*Storage tape*

$w_1$	$w_2$	$w_3$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-------	-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Storage tape*

<i>Simulation of TM <math>M_2</math> for string <math>w_3</math></i>										$B$	$B$
--	--	--	--	--	--	--	--	--	--	-----	-----

*Output tape*

#	$w_2$	#	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---	-------	---	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Generating Languages Accepted by a Turing Machine

*Storage tape*

$w_1$	$w_2$	$w_3$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-------	-------	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Storage tape*

<i>Simulation of TM <math>M_2</math> for string <math>w_3</math></i>										$B$	$B$
--	--	--	--	--	--	--	--	--	--	-----	-----

*Output tape*

#	$w_2$	#	$w_3$	#	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---	-------	---	-------	---	-----	-----	-----	-----	-----	-----	-----	-----

# Generating Languages Accepted by a Turing Machine

*Storage tape*

$w_1$	$w_2$	$w_3$	$w_4$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-------	-------	-------	-----	-----	-----	-----	-----	-----	-----

*Storage tape*

<i>Simulation of TM <math>M_2</math> for string <math>w_3</math></i>									$B$	$B$
--	--	--	--	--	--	--	--	--	-----	-----

*Output tape*

#	$w_2$	#	$w_3$	#	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---	-------	---	-------	---	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Generating Languages Accepted by a Turing Machine

*Storage tape*

$w_1$	$w_2$	$w_3$	$w_4$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-------	-------	-------	-----	-----	-----	-----	-----	-----	-----

*Storage tape*

<i>Simulation of TM <math>M_2</math> for string <math>w_4</math></i>									$B$	$B$
--	--	--	--	--	--	--	--	--	-----	-----

*Output tape*

#	$w_2$	#	$w_3$	#	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---	-------	---	-------	---	-----	-----	-----	-----	-----	-----	-----	-----

# Generating Languages Accepted by a Turing Machine

*Storage tape*

$w_1$	$w_2$	$w_3$	$w_4$	$w_5$	$B$	$B$	$B$	$B$	$B$
-------	-------	-------	-------	-------	-----	-----	-----	-----	-----

*Storage tape*

<i>Simulation of TM <math>M_2</math> for string <math>w_4</math></i>								$B$	$B$
--	--	--	--	--	--	--	--	-----	-----

*Output tape*

#	$w_2$	#	$w_3$	#	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---	-------	---	-------	---	-----	-----	-----	-----	-----	-----	-----	-----	-----



# Generating Languages Accepted by a Turing Machine

*Storage tape*

$w_1$	$w_2$	$w_3$	$w_4$	$w_5$	$B$	$B$	$B$	$B$	$B$
-------	-------	-------	-------	-------	-----	-----	-----	-----	-----

*Storage tape*

<i>Simulation of TM <math>M_2</math> for string <math>w_5</math></i>								$B$	$B$
--	--	--	--	--	--	--	--	-----	-----

*Output tape*

#	$w_2$	#	$w_3$	#	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---	-------	---	-------	---	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Generating Languages Accepted by a Turing Machine

*Storage tape*

$w_1$	$w_2$	$w_3$	$w_4$	$w_5$	$w_6$	$B$	$B$	$B$
-------	-------	-------	-------	-------	-------	-----	-----	-----

*Storage tape*

<i>Simulation of TM <math>M_2</math> for string <math>w_5</math></i>							$B$	$B$
--	--	--	--	--	--	--	-----	-----

*Output tape*

#	$w_2$	#	$w_3$	#	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---	-------	---	-------	---	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Generating Languages Accepted by a Turing Machine

*Storage tape*

$w_1$	$w_2$	$w_3$	$w_4$	$w_5$	$w_6$	$B$	$B$	$B$
-------	-------	-------	-------	-------	-------	-----	-----	-----

*Storage tape*

<i>Simulation of TM <math>M_2</math> for string <math>w_6</math></i>							$B$	$B$
--	--	--	--	--	--	--	-----	-----

*Output tape*

#	$w_2$	#	$w_3$	#	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---	-------	---	-------	---	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Generating Languages Accepted by a Turing Machine

*Storage tape*

$w_1$	$w_2$	$w_3$	$w_4$	$w_5$	$w_6$	$B$	$B$	$B$
-------	-------	-------	-------	-------	-------	-----	-----	-----

*Storage tape*

<i>Simulation of TM <math>M_2</math> for string <math>w_6</math></i>							$B$	$B$
--	--	--	--	--	--	--	-----	-----

*Output tape*

#	$w_2$	#	$w_3$	#	$w_6$	#	$B$	$B$	$B$	$B$	$B$	$B$
---	-------	---	-------	---	-------	---	-----	-----	-----	-----	-----	-----

# Generating Languages Accepted by a Turing Machine

*Storage tape*

$w_1$	$w_2$	$w_3$	$w_4$	$w_5$	$w_6$	$w_7$	$B$
-------	-------	-------	-------	-------	-------	-------	-----

*Storage tape*

<i>Simulation of TM <math>M_2</math> for string <math>w_6</math></i>							$B$	$B$
--	--	--	--	--	--	--	-----	-----

*Output tape*

#	$w_2$	#	$w_3$	#	$w_6$	#	$B$	$B$	$B$	$B$	$B$	$B$
---	-------	---	-------	---	-------	---	-----	-----	-----	-----	-----	-----

# Generating Languages Accepted by a Turing Machine

**Storage tape**

$w_1$	$w_2$	$w_3$	$w_4$	$w_5$	$w_6$	$w_7$	$B$
-------	-------	-------	-------	-------	-------	-------	-----

**Storage tape**

<i>Simulation of TM <math>M_2</math> for string <math>w_7</math></i>							$B$	$B$
--	--	--	--	--	--	--	-----	-----

**Output tape**

#	$w_2$	#	$w_3$	#	$w_6$	#	$B$	$B$	$B$	$B$	$B$	$B$
---	-------	---	-------	---	-------	---	-----	-----	-----	-----	-----	-----

# Generating Languages Accepted by a Turing Machine

**Storage tape**

$w_1$	$w_2$	$w_3$	$w_4$	$w_5$	$w_6$	$w_7$	$B$
-------	-------	-------	-------	-------	-------	-------	-----

**Storage tape**

<i>Simulation of TM <math>M_2</math> for string <math>w_7</math></i>							$B$	$B$
--	--	--	--	--	--	--	-----	-----

**Output tape**

#	$w_2$	#	$w_3$	#	$w_6$	#	$B$	$B$	$B$	$B$	$B$	$B$
---	-------	---	-------	---	-------	---	-----	-----	-----	-----	-----	-----

***Possible only for recursive languages***

# Generating Languages Accepted by a Turing Machine

*Storage tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

*Storage tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

*Storage tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

*Output tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------



# Generating Languages Accepted by a Turing Machine

## *Recursive enumerable languages*

*Storage tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

*Storage tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

*Storage tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

*Output tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

# Generating Languages Accepted by a Turing Machine

## *Recursive enumerable languages*

*Storage tape*

<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Storage tape*

<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Storage tape*

<i>( i, j )</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>
-----------------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Output tape*

<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

# Generating Languages Accepted by a Turing Machine

## *Recursive enumerable languages*

**Storage tape**

$w_i$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Storage tape**

$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Storage tape**

$(i, j)$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
----------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Output tape**

$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Generating Languages Accepted by a Turing Machine

## *Recursive enumerable languages*

**Storage tape**

$w_i$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Storage tape**

<i>Simulation of TM <math>M_2</math> for string <math>w_i</math> in <math>j</math> steps</i>	$B$	$B$
--	-----	-----

**Storage tape**

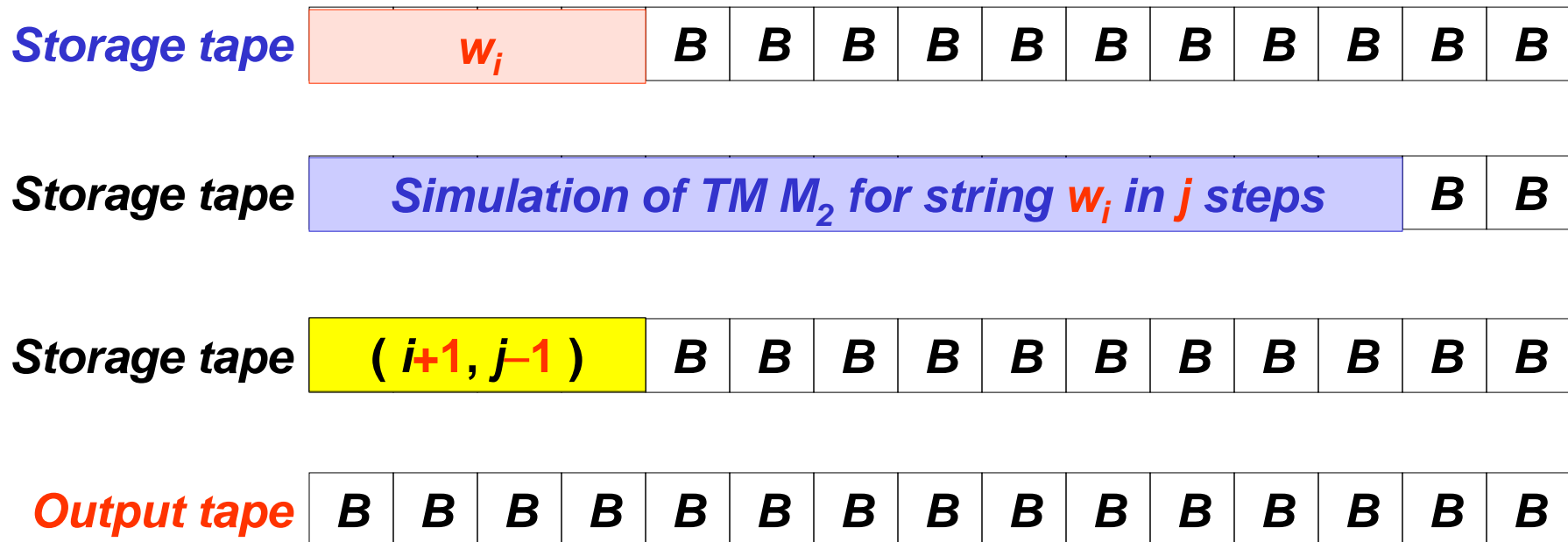
$(i, j)$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
----------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Output tape**

$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Generating Languages Accepted by a Turing Machine

## *Recursive enumerable languages*



# Generating Languages Accepted by a Turing Machine

## *Recursive enumerable languages*

**Storage tape**

$w_{i+1}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Storage tape**

<i>Simulation of TM <math>M_2</math> for string <math>w_i</math> in <math>j</math> steps</i>											$B$	$B$
--	--	--	--	--	--	--	--	--	--	--	-----	-----

**Storage tape**

$(i+1, j-1)$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
--------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Output tape**

$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Generating Languages Accepted by a Turing Machine

## *Recursive enumerable languages*

**Storage tape**

$w_{i+1}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Storage tape**

Simulation of TM $M_2$ for string $w_{i+1}$ in $j-1$ steps	$B$	$B$
--	-----	-----

**Storage tape**

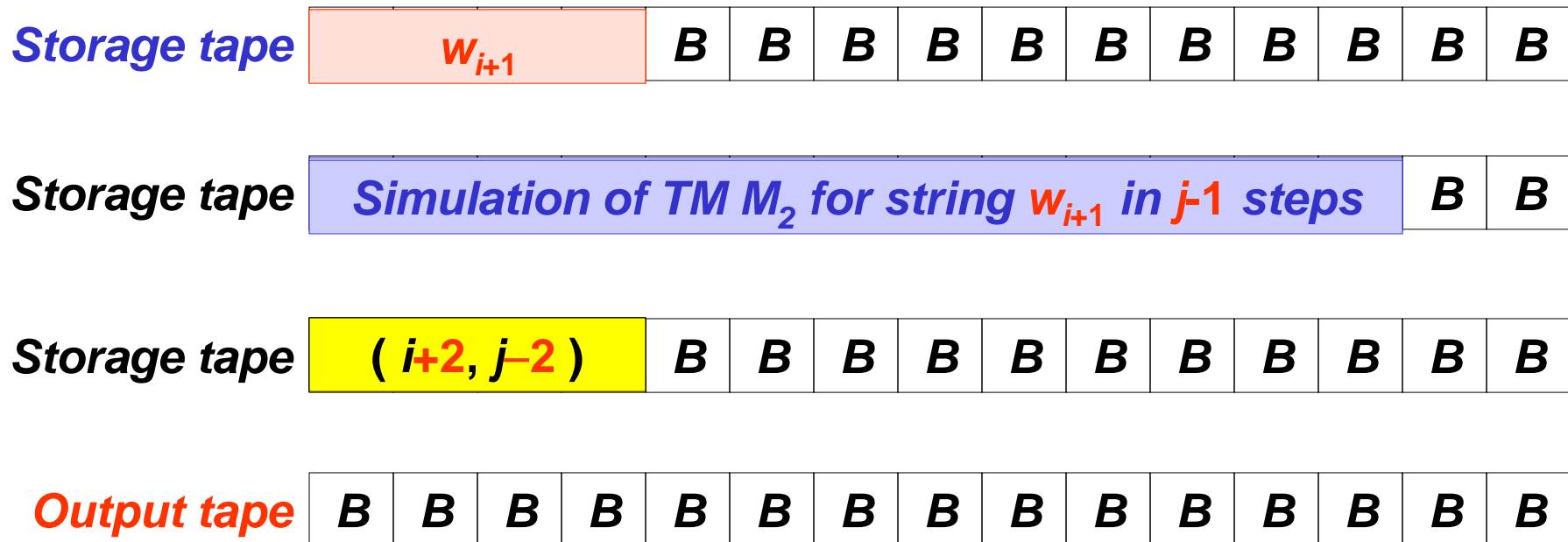
$(i+1, j-1)$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
--------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Output tape**

$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Generating Languages Accepted by a Turing Machine

## *Recursive enumerable languages*





# Generating Languages Accepted by a Turing Machine

## *Recursive enumerable languages*

**Storage tape**

$w_{i+2}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Storage tape**

Simulation of TM $M_2$ for string $w_{i+1}$ in $j-1$ steps	$B$	$B$
--	-----	-----

**Storage tape**

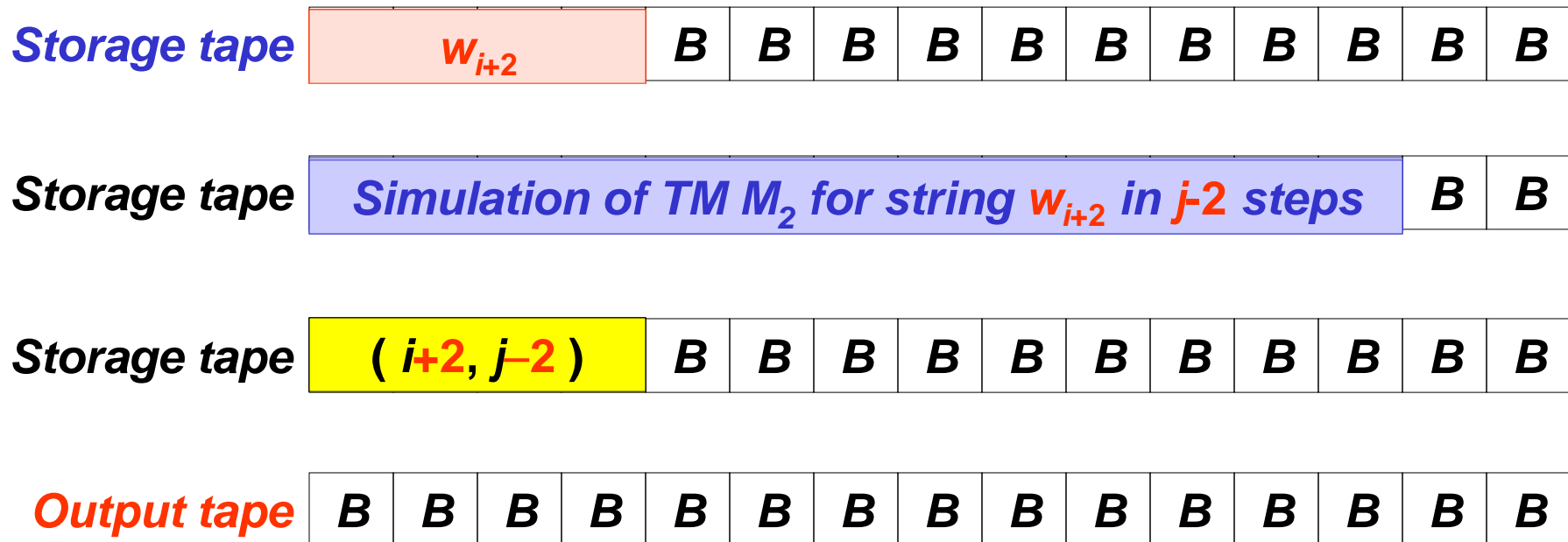
$(i+2, j-2)$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
--------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Output tape**

$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

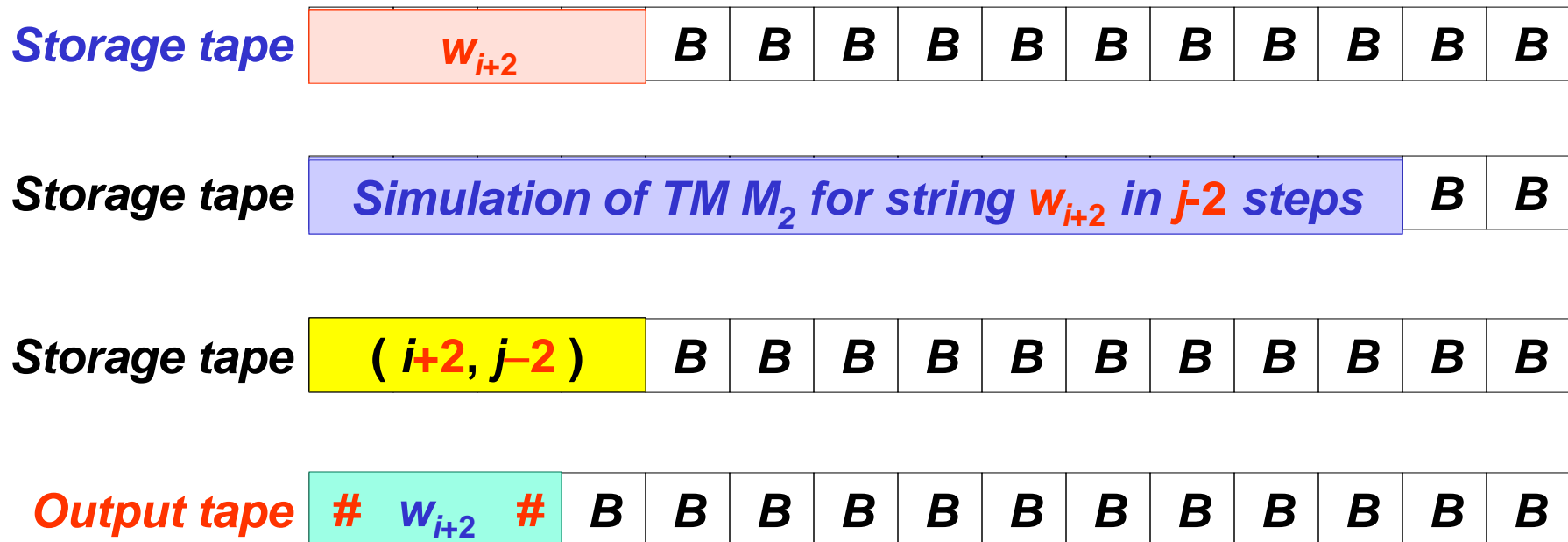
# Generating Languages Accepted by a Turing Machine

## *Recursive enumerable languages*



# Generating Languages Accepted by a Turing Machine

## *Recursive enumerable languages*



# Generating Languages Accepted by a Turing Machine

## *Recursive enumerable languages*

**Storage tape**

$w_{i+2}$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Storage tape**

Simulation of TM $M_2$ for string $w_{i+2}$ in $j-2$ steps	$B$	$B$
--	-----	-----

**Storage tape**

$(1, 1), 1+1=2$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Output tape**

# $w_{i+2}$ #	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

$(1, 1)$

# Generating Languages Accepted by a Turing Machine

## *Recursive enumerable languages*

**Storage tape**

$w_1$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Storage tape**

Simulation of TM $M_2$ for string $w_{i+2}$ in $j-2$ steps	$B$	$B$
--	-----	-----

**Storage tape**

$(1, 1), 1+1=2$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Output tape**

$\# w_{i+2} \#$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

$(1, 1)$

# Generating Languages Accepted by a Turing Machine

## *Recursive enumerable languages*

**Storage tape**

$w_1$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Storage tape**

<i>Simulation of TM <math>M_2</math> for string <math>w_1</math> in 1 step</i>										$B$	$B$
--	--	--	--	--	--	--	--	--	--	-----	-----

**Storage tape**

$(1, 1), 1+1=2$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Output tape**

$\#$	$w_{i+2}$	$\#$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
------	-----------	------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

$(1, 1)$

# Generating Languages Accepted by a Turing Machine

## *Recursive enumerable languages*

**Storage tape**

$w_1$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Storage tape**

Simulation of TM $M_2$ for string $w_1$ in 1 step	$B$	$B$
---	-----	-----

**Storage tape**

$(1, 2), 1+2=3$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Output tape**

# $w_{i+2}$ #	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

$(1,1), (1,2)$

# Generating Languages Accepted by a Turing Machine

## *Recursive enumerable languages*

**Storage tape**

$w_1$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Storage tape**

Simulation of TM $M_2$ for string $w_1$ in 1 step	$B$	$B$
---	-----	-----

**Storage tape**

$(1, 2), 1+2=3$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Output tape**

# $w_{i+2}$ #	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

$(1,1), (1,2)$



# Generating Languages Accepted by a Turing Machine

## *Recursive enumerable languages*

**Storage tape**

$w_1$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Storage tape**

Simulation of TM $M_2$ for string $w_1$ in 2 steps	$B$	$B$
--	-----	-----

**Storage tape**

( 1, 2 ), 1+2=3	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Output tape**

# $w_{i+2}$ #	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

(1,1), (1,2)

# Generating Languages Accepted by a Turing Machine

## *Recursive enumerable languages*

**Storage tape**

$w_1$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Storage tape**

Simulation of TM $M_2$ for string $w_1$ in 2 steps	$B$	$B$
--	-----	-----

**Storage tape**

$(2, 1), 2+1=3$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Output tape**

# $w_{i+2}$ #	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

$(1,1), (1,2), (2,1)$

# Generating Languages Accepted by a Turing Machine

## *Recursive enumerable languages*

Storage tape

$w_2$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Storage tape

Simulation of TM $M_2$ for string $w_1$ in 2 steps										$B$	$B$
--	--	--	--	--	--	--	--	--	--	-----	-----

Storage tape

$(2, 1), 2+1=3$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Output tape

# $w_{i+2}$ #	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

$(1,1), (1,2), (2,1)$

# Generating Languages Accepted by a Turing Machine

## *Recursive enumerable languages*

**Storage tape**

$w_2$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Storage tape**

Simulation of TM $M_2$ for string $w_2$ in 1 step										$B$	$B$
---	--	--	--	--	--	--	--	--	--	-----	-----

**Storage tape**

$(2, 1), 2+1=3$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Output tape**

#	$w_{i+2}$	#	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---	-----------	---	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

$(1,1), (1,2), (2,1)$

# Generating Languages Accepted by a Turing Machine

## *Recursive enumerable languages*

**Storage tape**

$w_2$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Storage tape**

Simulation of TM $M_2$ for string $w_2$ in 1 step										$B$	$B$
---	--	--	--	--	--	--	--	--	--	-----	-----

**Storage tape**

$(1, 3), 1+3=4$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Output tape**

#	$w_{i+2}$	#	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---	-----------	---	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

$(1,1), (1,2), (2,1), (1,3)$

# Generating Languages Accepted by a Turing Machine

## *Recursive enumerable languages*

**Storage tape**

$w_1$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Storage tape**

Simulation of TM $M_2$ for string $w_2$ in 1 step	$B$	$B$
---	-----	-----

**Storage tape**

$(1, 3), 1+3=4$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Output tape**

# $w_{i+2}$ #	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

$(1,1), (1,2), (2,1), (1,3)$

# Generating Languages Accepted by a Turing Machine

## *Recursive enumerable languages*

**Storage tape**

$w_1$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Storage tape**

Simulation of TM $M_2$ for string $w_1$ in 3 steps										$B$	$B$
--	--	--	--	--	--	--	--	--	--	-----	-----

**Storage tape**

$(1, 3), 1+3=4$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Output tape**

#	$w_{i+2}$	#	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---	-----------	---	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

$(1,1), (1,2), (2,1), (1,3)$

# Generating Languages Accepted by a Turing Machine

## *Recursive enumerable languages*

**Storage tape**

$w_1$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Storage tape**

Simulation of TM $M_2$ for string $w_1$ in 3 steps										$B$	$B$
--	--	--	--	--	--	--	--	--	--	-----	-----

**Storage tape**

$(2, 2), 2+2=4$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Output tape**

$\#$	$w_{i+2}$	$\#$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
------	-----------	------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

$(1,1), (1,2), (2,1), (1,3), (2,2)$



# Generating Languages Accepted by a Turing Machine

## *Recursive enumerable languages*

**Storage tape**

$w_2$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Storage tape**

Simulation of TM $M_2$ for string $w_1$ in 3 steps	$B$	$B$
--	-----	-----

**Storage tape**

$(2, 2), 2+2=4$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Output tape**

# $w_{i+2}$ #	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

$(1,1), (1,2), (2,1), (1,3), (2,2)$

# Generating Languages Accepted by a Turing Machine

## *Recursive enumerable languages*

**Storage tape**

$w_2$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Storage tape**

Simulation of TM $M_2$ for string $w_2$ in 2 steps	$B$	$B$
--	-----	-----

**Storage tape**

$(2, 2), 2+2=4$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Output tape**

# $w_{i+2}$ #	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

$(1,1), (1,2), (2,1), (1,3), (2,2)$

# Generating Languages Accepted by a Turing Machine

## *Recursive enumerable languages*

**Storage tape**

$w_2$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Storage tape**

Simulation of TM $M_2$ for string $w_2$ in 2 steps	$B$	$B$
--	-----	-----

**Storage tape**

$(2, 2), 2+2=4$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Output tape**

# $w_{i+2}$ #	$w_2$ #	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------------	---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

$(1,1), (1,2), (2,1), (1,3), (2,2)$

# Generating Languages Accepted by a Turing Machine

## *Recursive enumerable languages*

**Storage tape**

$w_2$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Storage tape**

Simulation of TM $M_2$ for string $w_2$ in 2 steps	$B$	$B$
--	-----	-----

**Storage tape**

$(3, 1), 3+1=4$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Output tape**

# $w_{i+2}$ #	$w_2$ #	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------------	---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

$(1,1), (1,2), (2,1), (1,3), (2,2), (3,1)$

# Generating Languages Accepted by a Turing Machine

## *Recursive enumerable languages*

**Storage tape**

$w_3$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Storage tape**

Simulation of TM $M_2$ for string $w_2$ in 2 steps	$B$	$B$
--	-----	-----

**Storage tape**

$(3, 1), 3+1=4$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**Output tape**

# $w_{i+2}$ #	$w_2$ #	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------------	---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

$(1,1), (1,2), (2,1), (1,3), (2,2), (3,1)$

# Generating Languages Accepted by a Turing Machine

## *Recursive enumerable languages*

Storage tape 

$w_3$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Storage tape 

Simulation of TM $M_2$ for string $w_3$ in 1 one	$B$	$B$
--	-----	-----

Storage tape 

$(3, 1), 3+1=4$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Output tape 

# $w_{i+2}$ #	$w_2$ #	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------------	---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

$(1,1), (1,2), (2,1), (1,3), (2,2), (3,1)$

# Generating Languages Accepted by a Turing Machine

## *Recursive enumerable languages*

Storage tape 

$w_3$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Storage tape 

Simulation of TM $M_2$ for string $w_3$ in 1 one	$B$	$B$
--	-----	-----

Storage tape 

$(3, 1), 3+1=4$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Output tape 

#	$w_{i+2}$	#	$w_2$	#	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---	-----------	---	-------	---	-----	-----	-----	-----	-----	-----	-----	-----	-----

$(1,1), (1,2), (2,1), (1,3), (2,2), (3,1), \dots$

**It is possible to generate recursive languages in canonical ordering**



**It is possible to generate recursive languages in canonical ordering**

**Binary alphabet:**       $\Sigma = \{0, 1\}$

# It is possible to generate recursive languages in canonical ordering

**Binary alphabet:**  $\Sigma = \{0, 1\}$

**Canonical ordering:**  $\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011,$

# It is possible to generate recursive languages in canonical ordering

**Binary alphabet:**  $\Sigma = \{0, 1\}$

**Canonical ordering:**  $\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, 0000, 0001, \dots$

# It is possible to generate recursive languages in canonical ordering

Binary alphabet:  $\Sigma = \{0, 1\}$

Canonical ordering:  $\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, 0000, 0001, \dots$

*Storage tape*

<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Storage tape*

<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Output tape*

<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

# It is possible to generate recursive languages in canonical ordering

Binary alphabet:  $\Sigma = \{0, 1\}$

Canonical ordering:  $\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, 0000, 0001, \dots$

*Storage tape*

$\epsilon$	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>
------------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Storage tape*

<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Output tape*

<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

# It is possible to generate recursive languages in canonical ordering

Binary alphabet:  $\Sigma = \{0, 1\}$

Canonical ordering:  $\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, 0000, 0001, \dots$

*Storage tape*

$\varepsilon$	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>
---------------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Storage tape*

<i>Simulation of TM <math>M_2</math> for string <math>\varepsilon</math></i>											<i>B</i>	<i>B</i>
--	--	--	--	--	--	--	--	--	--	--	----------	----------

*Output tape*

<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

# It is possible to generate recursive languages in canonical ordering

Binary alphabet:  $\Sigma = \{0, 1\}$

Canonical ordering:  $\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, 0000, 0001, \dots$

Storage tape

$\epsilon$	0	B	B	B	B	B	B	B	B	B	B	B
------------	---	---	---	---	---	---	---	---	---	---	---	---

Storage tape

Simulation of TM $M_2$ for string $\epsilon$											B	B
--	--	--	--	--	--	--	--	--	--	--	---	---

Output tape

B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

# It is possible to generate recursive languages in canonical ordering

Binary alphabet:  $\Sigma = \{0, 1\}$

Canonical ordering:  $\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, 0000, 0001, \dots$

*Storage tape*

$\epsilon$	0	B	B	B	B	B	B	B	B	B	B	B
------------	---	---	---	---	---	---	---	---	---	---	---	---

*Storage tape*

<i>Simulation of TM <math>M_2</math> for string 0</i>											B	B
---	--	--	--	--	--	--	--	--	--	--	---	---

*Output tape*

B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---



# It is possible to generate recursive languages in canonical ordering

Binary alphabet:  $\Sigma = \{0, 1\}$

Canonical ordering:  $\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, 0000, 0001, \dots$

Storage tape

$\epsilon$	0	B	B	B	B	B	B	B	B	B	B	B
------------	---	---	---	---	---	---	---	---	---	---	---	---

Storage tape

Simulation of TM $M_2$ for string 0										B	B
-------------------------------------	--	--	--	--	--	--	--	--	--	---	---

Output tape

#	0	#	B	B	B	B	B	B	B	B	B	B	B
---	---	---	---	---	---	---	---	---	---	---	---	---	---

# It is possible to generate recursive languages in canonical ordering

Binary alphabet:  $\Sigma = \{0, 1\}$

Canonical ordering:  $\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, 0000, 0001, \dots$

Storage tape

$\epsilon$	0	1	B	B	B	B	B	B	B	B	B
------------	---	---	---	---	---	---	---	---	---	---	---

Storage tape

Simulation of TM $M_2$ for string 0										B	B
-------------------------------------	--	--	--	--	--	--	--	--	--	---	---

Output tape

#	0	#	B	B	B	B	B	B	B	B	B	B	B
---	---	---	---	---	---	---	---	---	---	---	---	---	---

# It is possible to generate recursive languages in canonical ordering

Binary alphabet:  $\Sigma = \{0, 1\}$

Canonical ordering:  $\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, 0000, 0001, \dots$

Storage tape

$\epsilon$	0	1	B	B	B	B	B	B	B	B	B
------------	---	---	---	---	---	---	---	---	---	---	---

Storage tape

Simulation of TM $M_2$ for string 1										B	B
-------------------------------------	--	--	--	--	--	--	--	--	--	---	---

Output tape

#	0	#	B	B	B	B	B	B	B	B	B	B	B
---	---	---	---	---	---	---	---	---	---	---	---	---	---

# It is possible to generate recursive languages in canonical ordering

Binary alphabet:  $\Sigma = \{0, 1\}$

Canonical ordering:  $\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, 0000, 0001, \dots$

**Storage tape**

$\epsilon$	0	1	B	B	B	B	B	B	B	B	B
------------	---	---	---	---	---	---	---	---	---	---	---

**Storage tape**

<i>Simulation of TM <math>M_2</math> for string 1</i>										B	B
---	--	--	--	--	--	--	--	--	--	---	---

**Output tape**

#	0	#	1	#	B	B	B	B	B	B	B	B
---	---	---	---	---	---	---	---	---	---	---	---	---

# It is possible to generate recursive languages in canonical ordering

Binary alphabet:  $\Sigma = \{0, 1\}$

Canonical ordering:  $\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, 0000, 0001, \dots$

*Storage tape*

$\epsilon$	0	1	00	B	B	B	B	B	B	B
------------	---	---	----	---	---	---	---	---	---	---

*Storage tape*

<i>Simulation of TM <math>M_2</math> for string 1</i>									B	B
---	--	--	--	--	--	--	--	--	---	---

*Output tape*

#	0	#	1	#	B	B	B	B	B	B	B	B
---	---	---	---	---	---	---	---	---	---	---	---	---

# It is possible to generate recursive languages in canonical ordering

Binary alphabet:  $\Sigma = \{0, 1\}$

Canonical ordering:  $\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, 0000, 0001, \dots$

**Storage tape**

$\epsilon$	0	1	00	B	B	B	B	B	B	B
------------	---	---	----	---	---	---	---	---	---	---

**Storage tape**

<i>Simulation of TM <math>M_2</math> for string 00</i>									B	B
--	--	--	--	--	--	--	--	--	---	---

**Output tape**

#	0	#	1	#	B	B	B	B	B	B	B	B
---	---	---	---	---	---	---	---	---	---	---	---	---

# It is possible to generate recursive languages in canonical ordering

Binary alphabet:  $\Sigma = \{0, 1\}$

Canonical ordering:  $\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, 0000, 0001, \dots$

**Storage tape**

$\varepsilon$	0	1	00	01	B	B	B	B	B
---------------	---	---	----	----	---	---	---	---	---

**Storage tape**

<i>Simulation of TM <math>M_2</math> for string 00</i>								B	B
--	--	--	--	--	--	--	--	---	---

**Output tape**

#	0	#	1	#	B	B	B	B	B	B	B	B	B
---	---	---	---	---	---	---	---	---	---	---	---	---	---

# It is possible to generate recursive languages in canonical ordering

Binary alphabet:  $\Sigma = \{0, 1\}$

Canonical ordering:  $\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, 0000, 0001, \dots$

**Storage tape**

$\epsilon$	0	1	00	01	B	B	B	B	B
------------	---	---	----	----	---	---	---	---	---

**Storage tape**

<i>Simulation of TM <math>M_2</math> for string 01</i>								B	B
--	--	--	--	--	--	--	--	---	---

**Output tape**

#	0	#	1	#	B	B	B	B	B	B	B	B	B
---	---	---	---	---	---	---	---	---	---	---	---	---	---



# It is possible to generate recursive languages in canonical ordering

Binary alphabet:  $\Sigma = \{0, 1\}$

Canonical ordering:  $\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, 0000, 0001, \dots$

<i>Storage tape</i>	$\epsilon$	0	1	00	01	10	B	B	B
---------------------	------------	---	---	----	----	----	---	---	---

<i>Storage tape</i>	<i>Simulation of TM <math>M_2</math> for string 01</i>							B	B
---------------------	--	--	--	--	--	--	--	---	---

<i>Output tape</i>	#	0	#	1	#	B	B	B	B	B	B	B	B
--------------------	---	---	---	---	---	---	---	---	---	---	---	---	---

# It is possible to generate recursive languages in canonical ordering

Binary alphabet:  $\Sigma = \{0, 1\}$

Canonical ordering:  $\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, 0000, 0001, \dots$

Storage tape

$\epsilon$	0	1	00	01	10	B	B	B
------------	---	---	----	----	----	---	---	---

Storage tape

<i>Simulation of TM <math>M_2</math> for string 10</i>							B	B
--	--	--	--	--	--	--	---	---

Output tape

#	0	#	1	#	B	B	B	B	B	B	B	B	B
---	---	---	---	---	---	---	---	---	---	---	---	---	---

# It is possible to generate recursive languages in canonical ordering

Binary alphabet:  $\Sigma = \{0, 1\}$

Canonical ordering:  $\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, 0000, 0001, \dots$

Storage tape

$\varepsilon$	0	1	00	01	10	B	B	B
---------------	---	---	----	----	----	---	---	---

Storage tape

<i>Simulation of TM <math>M_2</math> for string 10</i>							B	B
--	--	--	--	--	--	--	---	---

Output tape

#	0	#	1	#	10	#	B	B	B	B	B	B
---	---	---	---	---	----	---	---	---	---	---	---	---

# It is possible to generate recursive languages in canonical ordering

Binary alphabet:  $\Sigma = \{0, 1\}$

Canonical ordering:  $\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, 0000, 0001, \dots$

Storage tape

$\varepsilon$	0	1	00	01	10	11	B
---------------	---	---	----	----	----	----	---

Storage tape

<i>Simulation of TM <math>M_2</math> for string 10</i>							B	B
--	--	--	--	--	--	--	---	---

Output tape

#	0	#	1	#	10	#	B	B	B	B	B	B
---	---	---	---	---	----	---	---	---	---	---	---	---

# It is possible to generate recursive languages in canonical ordering

Binary alphabet:  $\Sigma = \{0, 1\}$

Canonical ordering:  $\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, 0000, 0001, \dots$

Storage tape	$\varepsilon$	0	1	00	01	10	11	B
--------------	---------------	---	---	----	----	----	----	---

Storage tape	<i>Simulation of TM <math>M_2</math> for string 11</i>							B	B
--------------	--	--	--	--	--	--	--	---	---

Output tape	#	0	#	1	#	10	#	B	B	B	B	B	B
-------------	---	---	---	---	---	----	---	---	---	---	---	---	---

**Language that is possible to generate in canonical ordering  
is a recursive language**

Language that is possible to generate in canonical ordering  
is a recursive language

*Input tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

*Storage tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

*Output tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

Language that is possible to generate in canonical ordering  
is a recursive language

*Input tape*

$w_i$	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>
-------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Storage tape*

<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

*Output tape*

<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------



Language that is possible to generate in canonical ordering  
is a recursive language

*Input tape*

$w_i$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Storage tape*

$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Output tape*

#	$w_1$	#	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---	-------	---	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Language that is possible to generate in canonical ordering  
is a recursive language

*Input tape*

$w_i$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Storage tape*

$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Output tape*

$\#w_i$	$w_1$	$\#$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-------	------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Language that is possible to generate in canonical ordering  
is a recursive language

*Input tape*

$w_i$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Storage tape*

$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Output tape*

#	$w_1$	#	$w_2$	#	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---	-------	---	-------	---	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Language that is possible to generate in canonical ordering  
is a recursive language

*Input tape*

$w_i$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Storage tape*

$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Output tape*

#	$w_1$	#	$w_2 w_i$	#	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---	-------	---	-----------	---	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Language that is possible to generate in canonical ordering  
is a recursive language

*Input tape*

$w_i$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Storage tape*

$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Output tape*

#	$w_1$	#	$w_2$	#	$w_3$	#	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---	-------	---	-------	---	-------	---	-----	-----	-----	-----	-----	-----	-----	-----

Language that is possible to generate in canonical ordering  
is a recursive language

*Input tape*

$w_i$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Storage tape*

$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Output tape*

#	$w_1$	#	$w_2$	#	$w_3 w_i$	#	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---	-------	---	-------	---	-----------	---	-----	-----	-----	-----	-----	-----	-----	-----

Language that is possible to generate in canonical ordering  
is a recursive language

*Input tape*

$w_i$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Storage tape*

$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Output tape*

#	$w_1$	#	$w_2$	#	$w_3$	#	$B$	$B$	$w_i$	#	$B$	$B$	$B$	$B$
---	-------	---	-------	---	-------	---	-----	-----	-------	---	-----	-----	-----	-----

Language that is possible to generate in canonical ordering  
is a recursive language

*Input tape*

$w_i$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Storage tape*

$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Output tape*

#	$w_1$	#	$w_2$	#	$w_3$	#	$B$	$B$	$w_i w_i$ #	$B$	$B$	$B$	$B$
---	-------	---	-------	---	-------	---	-----	-----	-------------	-----	-----	-----	-----



Language that is possible to generate in canonical ordering  
is a recursive language

*Input tape*

$w_i$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Storage tape*

$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Output tape*

#	$w_1$	#	$w_2$	#	$w_3$	#	$B$	$B$	$w_i$	#	$B$	$B$	$B$	$B$
---	-------	---	-------	---	-------	---	-----	-----	-------	---	-----	-----	-----	-----

***YES – the string is accepted***

**Language that is possible to generate in canonical ordering  
is a recursive language**

## Language that is possible to generate in canonical ordering is a recursive language

*Input tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

*Storage tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

*Output tape*

<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>	<b><i>B</i></b>
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

## Language that is possible to generate in canonical ordering is a recursive language

*Input tape*

$w_i$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Storage tape*

$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Output tape*

$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Language that is possible to generate in canonical ordering  
is a recursive language

*Input tape*

$w_i$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Storage tape*

$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Output tape*

#	$w_1$	#	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---	-------	---	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Language that is possible to generate in canonical ordering  
is a recursive language

*Input tape*

$w_i$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Storage tape*

$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Output tape*

$\#w_i$	$w_1$	$\#$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---------	-------	------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Language that is possible to generate in canonical ordering  
is a recursive language

*Input tape*

$w_i$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Storage tape*

$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Output tape*

#	$w_1$	#	$w_2$	#	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---	-------	---	-------	---	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

## Language that is possible to generate in canonical ordering is a recursive language

*Input tape*

$w_i$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Storage tape*

$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Output tape*

#	$w_1$	#	$w_2 w_i$	#	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---	-------	---	-----------	---	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----



## Language that is possible to generate in canonical ordering is a recursive language

*Input tape*

$w_i$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Storage tape*

$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Output tape*

#	$w_1$	#	$w_2$	#	$w_3$	#	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---	-------	---	-------	---	-------	---	-----	-----	-----	-----	-----	-----	-----	-----

## Language that is possible to generate in canonical ordering is a recursive language

*Input tape*

$w_i$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Storage tape*

$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Output tape*

#	$w_1$	#	$w_2$	#	$w_3 w_i$	#	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
---	-------	---	-------	---	-----------	---	-----	-----	-----	-----	-----	-----	-----	-----

Language that is possible to generate in canonical ordering  
is a recursive language

*Input tape*

$w_i$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Storage tape*

$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Output tape*

#	$w_1$	#	$w_2$	#	$w_3$	#	$B$	$B$	$w_j$	#	$B$	$B$	$B$	$B$
---	-------	---	-------	---	-------	---	-----	-----	-------	---	-----	-----	-----	-----

Language that is possible to generate in canonical ordering  
is a recursive language

*Input tape*

$w_i$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Storage tape*

$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Output tape*

#	$w_1$	#	$w_2$	#	$w_3$	#	$B$	$B$	$w_j w_i$ #	$B$	$B$	$B$	$B$
---	-------	---	-------	---	-------	---	-----	-----	-------------	-----	-----	-----	-----

Language that is possible to generate in canonical ordering  
is a recursive language

*Input tape*

$w_i$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Storage tape*

$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Output tape*

#	$w_1$	#	$w_2$	#	$w_3$	#	$B$	$B$	$w_j$	#	$B$	$B$	$B$	$B$
---	-------	---	-------	---	-------	---	-----	-----	-------	---	-----	-----	-----	-----

*In canonical ordering  $j$  is before  $i$*

Language that is possible to generate in canonical ordering  
is a recursive language

*Input tape*

$w_i$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Storage tape*

$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Output tape*

#	$w_1$	#	$w_2$	#	$w_3$	#	$B$	$B$	$w_j$	#	$B$	$B$	$B$	$B$
---	-------	---	-------	---	-------	---	-----	-----	-------	---	-----	-----	-----	-----

*In canonical ordering  $j$  is before  $i$*

**NO** – the string is not accepted

Language that is possible to generate in canonical ordering  
is a recursive language

*Input tape*

$w_i$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Storage tape*

$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$	$B$
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

*Output tape*

#	$w_1$	#	$w_2$	#	$w_3$	#	$B$	$B$	$w_j$	#	$B$	$B$	$B$	$B$
---	-------	---	-------	---	-------	---	-----	-----	-------	---	-----	-----	-----	-----

*In canonical ordering  $j$  is before  $i$*

**NO** – the string is not accepted

*If the language is finite*

*and if a string is processed that follows all the strings in  $L(M_2)$   
then TM  $M_2$  never stops*