Lecture 14

Lecture 14

Unrestricted Grammar

- **Productions of a context-free grammar**
  - only one nonterminal symbol (variable) on the left side

$$A \to \alpha$$

- **Productions of a regular grammar**
  - left-linear of right-linear

$$A \to wB \qquad\qquad A \to Bw$$
$$A \to w \qquad\qquad A \to w$$

- **Productions of an unrestricted grammar**
  - $\alpha \to \beta$

$$\alpha \neq \varepsilon$$
$$\gamma\,\alpha\,\delta \Rightarrow \gamma\,\beta\,\delta$$

Unrestricted Grammar

1) $S \to ACaB$
2) $Ca \to aaC$
3) $CB \to DB$
4) $CB \to E$
5) $aD \to Da$
6) $AD \to AC$
7) $aE \to Ea$
8) $AE \to \varepsilon$

Unrestricted Grammar

1) $S \to ACaB$
2) $Ca \to aaC$
3) $CB \to DB$
4) $CB \to E$
5) $aD \to Da$
6) $AD \to AC$
7) $aE \to Ea$
8) $AE \to \varepsilon$

S

Unrestricted Grammar

1) $S \rightarrow ACaB$
2) $Ca \rightarrow aaC$
3) $CB \rightarrow DB$
4) $CB \rightarrow E$
5) $aD \rightarrow Da$
6) $AD \rightarrow AC$
7) $aE \rightarrow Ea$
8) $AE \rightarrow \varepsilon$

S

Unrestricted Grammar

1) $S \rightarrow ACaB$
2) $Ca \rightarrow aaC$
3) $CB \rightarrow DB$
4) $CB \rightarrow E$
5) $aD \rightarrow Da$
6) $AD \rightarrow AC$
7) $aE \rightarrow Ea$
8) $AE \rightarrow \varepsilon$

$$A \quad Ca \quad B$$

Unrestricted Grammar

1) $S \rightarrow ACaB$
2) $Ca \rightarrow aaC$
3) $CB \rightarrow DB$
4) $CB \rightarrow E$
5) $aD \rightarrow Da$
6) $AD \rightarrow AC$
7) $aE \rightarrow Ea$
8) $AE \rightarrow \varepsilon$

$A \quad Ca \quad B$

Unrestricted Grammar

1) $S \rightarrow ACaB$
2) $Ca \rightarrow aaC$
3) $CB \rightarrow DB$
4) $CB \rightarrow E$
5) $aD \rightarrow Da$
6) $AD \rightarrow AC$
7) $aE \rightarrow Ea$
8) $AE \rightarrow \varepsilon$

$A \quad a \quad a\ CB$

Unrestricted Grammar

1) $S \to ACaB$
2) $Ca \to aaC$
3) $CB \to DB$
4) $CB \to E$
5) $aD \to Da$
6) $AD \to AC$
7) $aE \to Ea$
8) $AE \to \varepsilon$

$A \quad a \quad a\ CB$

Unrestricted Grammar

1) $S \rightarrow ACaB$
2) $Ca \rightarrow aaC$
3) $CB \rightarrow DB$
4) $CB \rightarrow E$
5) $aD \rightarrow Da$
6) $AD \rightarrow AC$
7) $aE \rightarrow Ea$
8) $AE \rightarrow \varepsilon$

$A \quad a \quad a \quad E$

Unrestricted Grammar

1) $S \rightarrow ACaB$
2) $Ca \rightarrow aaC$
3) $CB \rightarrow DB$
4) $CB \rightarrow E$
5) $aD \rightarrow Da$
6) $AD \rightarrow AC$
7) $aE \rightarrow Ea$
8) $AE \rightarrow \varepsilon$

$A \quad a \quad a \quad E$

Unrestricted Grammar

1) $S \to ACaB$
2) $Ca \to aaC$
3) $CB \to DB$
4) $CB \to E$
5) $aD \to Da$
6) $AD \to AC$
7) $aE \to Ea$
8) $AE \to \varepsilon$

$A \quad aEa$

Unrestricted Grammar

1) $S \rightarrow ACaB$
2) $Ca \rightarrow aaC$
3) $CB \rightarrow DB$
4) $CB \rightarrow E$
5) $aD \rightarrow Da$
6) $AD \rightarrow AC$
7) $aE \rightarrow Ea$
8) $AE \rightarrow \varepsilon$

$AEa \quad a$

Unrestricted Grammar

1) $S \to ACaB$
2) $Ca \to aaC$
3) $CB \to DB$
4) $CB \to E$
5) $aD \to Da$
6) $AD \to AC$
7) $aE \to Ea$
8) $AE \to \varepsilon$

$AEa\ \ a$

Unrestricted Grammar

1) $S \to ACaB$
2) $Ca \to aaC$
3) $CB \to DB$
4) $CB \to E$
5) $aD \to Da$
6) $AD \to AC$
7) $aE \to Ea$
8) $AE \to \varepsilon$

$a \quad a$

Unrestricted Grammar

1) $S \rightarrow ACaB$
2) $Ca \rightarrow aaC$
3) $CB \rightarrow DB$
4) $CB \rightarrow E$
5) $aD \rightarrow Da$
6) $AD \rightarrow AC$
7) $aE \rightarrow Ea$
8) $AE \rightarrow \varepsilon$

$a \quad a$

$\boxed{S}$

Unrestricted Grammar

1) $S \rightarrow ACaB$
2) $Ca \rightarrow aaC$
3) $CB \rightarrow DB$
4) $CB \rightarrow E$
5) $aD \rightarrow Da$
6) $AD \rightarrow AC$
7) $aE \rightarrow Ea$
8) $AE \rightarrow \varepsilon$

$a \quad a$

$A\ C\ a\ B$

Unrestricted Grammar

1) $S \rightarrow ACaB$
2) $Ca \rightarrow aaC$
3) $CB \rightarrow DB$
4) $CB \rightarrow E$
5) $aD \rightarrow Da$
6) $AD \rightarrow AC$
7) $aE \rightarrow Ea$
8) $AE \rightarrow \varepsilon$

$a \quad a$

$A\ a\ a\ C\ B$

Unrestricted Grammar

1) $S \rightarrow ACaB$
2) $Ca \rightarrow aaC$
3) $CB \rightarrow DB$
4) $CB \rightarrow E$
5) $aD \rightarrow Da$
6) $AD \rightarrow AC$
7) $aE \rightarrow Ea$
8) $AE \rightarrow \varepsilon$

$A\ a\ a\ D\ B$

$a \quad a$

Unrestricted Grammar

1) $S \rightarrow ACaB$
2) $Ca \rightarrow aaC$
3) $CB \rightarrow DB$
4) $CB \rightarrow E$
5) $aD \rightarrow Da$
6) $AD \rightarrow AC$
7) $aE \rightarrow Ea$
8) $AE \rightarrow \varepsilon$

a *a*

A a D a B

Unrestricted Grammar

1) $S \rightarrow ACaB$
2) $Ca \rightarrow aaC$
3) $CB \rightarrow DB$
4) $CB \rightarrow E$
5) $aD \rightarrow Da$
6) $AD \rightarrow AC$
7) $aE \rightarrow Ea$
8) $AE \rightarrow \varepsilon$

a a

A D a a B

Unrestricted Grammar

1) $S \to ACaB$
2) $Ca \to aaC$
3) $CB \to DB$
4) $CB \to E$
5) $aD \to Da$
6) $AD \to AC$
7) $aE \to Ea$
8) $AE \to \varepsilon$

$a \quad a$

$A\ C\ a\ a\ B$

Unrestricted Grammar

1) $S \to ACaB$
2) $Ca \to aaC$
3) $CB \to DB$
4) $CB \to E$
5) $aD \to Da$
6) $AD \to AC$
7) $aE \to Ea$
8) $AE \to \varepsilon$

$a \quad a$

$A\ a\ a\ C\ a\ B$

Unrestricted Grammar

1) $S \rightarrow ACaB$
2) $Ca \rightarrow aaC$
3) $CB \rightarrow DB$
4) $CB \rightarrow E$
5) $aD \rightarrow Da$
6) $AD \rightarrow AC$
7) $aE \rightarrow Ea$
8) $AE \rightarrow \varepsilon$

$a \quad a$

$\boxed{A\ a\ a\ a\ a\ C\ B}$

Unrestricted Grammar

1) $S \to ACaB$
2) $Ca \to aaC$
3) $CB \to DB$
4) $CB \to E$
5) $aD \to Da$
6) $AD \to AC$
7) $aE \to Ea$
8) $AE \to \varepsilon$

$a \quad a$

$A\,a\,a\,a\,a\,E$

Unrestricted Grammar

1) $S \to ACaB$
2) $Ca \to aaC$
3) $CB \to DB$
4) $CB \to E$
5) $aD \to Da$
6) $AD \to AC$
7) $aE \to Ea$
8) $AE \to \varepsilon$

$a \quad a$

$A\ a\ a\ a\ E\ a$

Unrestricted Grammar

1) $S \rightarrow ACaB$

2) $Ca \rightarrow aaC$

3) $CB \rightarrow DB$

4) $CB \rightarrow E$

5) $aD \rightarrow Da$

6) $AD \rightarrow AC$

7) $aE \rightarrow Ea$

8) $AE \rightarrow \varepsilon$

$a \quad a$

$A\,a\,a\,E\,a\,a$

Unrestricted Grammar

1) $S \rightarrow ACaB$
2) $Ca \rightarrow aaC$
3) $CB \rightarrow DB$
4) $CB \rightarrow E$
5) $aD \rightarrow Da$
6) $AD \rightarrow AC$
7) $aE \rightarrow Ea$
8) $AE \rightarrow \varepsilon$

$a \quad a$

$A\ a\ E\ a\ a\ a$

Unrestricted Grammar

1) $S \rightarrow ACaB$
2) $Ca \rightarrow aaC$
3) $CB \rightarrow DB$
4) $CB \rightarrow E$
5) $aD \rightarrow Da$
6) $AD \rightarrow AC$
7) $aE \rightarrow Ea$
8) $AE \rightarrow \varepsilon$

$a \quad a$

$A\ E\ a\ a\ a\ a$

Unrestricted Grammar

1) $S \rightarrow ACaB$
2) $Ca \rightarrow aaC$
3) $CB \rightarrow DB$
4) $CB \rightarrow E$
5) $aD \rightarrow Da$
6) $AD \rightarrow AC$
7) $aE \rightarrow Ea$
8) $AE \rightarrow \varepsilon$

a *a*

a a a a

Unrestricted Grammar

1) $S \rightarrow ACaB$
2) $Ca \rightarrow aaC$
3) $CB \rightarrow DB$
4) $CB \rightarrow E$
5) $aD \rightarrow Da$
6) $AD \rightarrow AC$
7) $aE \rightarrow Ea$
8) $AE \rightarrow \varepsilon$

$a \quad a$

$a\ a\ a\ a$

S

$A\ a^i\ C\ a^j\ B$ $\quad i + 2j = 2^k$

$A\ a^i\ D\ B$ $\quad i = 2^k$

$A\ a^j\ D\ a^l\ B$ $\quad j + l = 2^k$

$A\ a^i\ E$ $\quad i = 2^k$

$A\ E\ a^i$ $\quad i = 2^k$

a^i $\quad i = 2^k$

Lecture 14

Construction of a Turing Machine from an Unrestricted Grammar

Tape	*w*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*

Tape	*S*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*

Construction of a Turing Machine from an Unrestricted Grammar

Tape	***w***	***B***	***B***	***B***	***B***	***B***	***B***	***B***	***B***	***B***	***B***

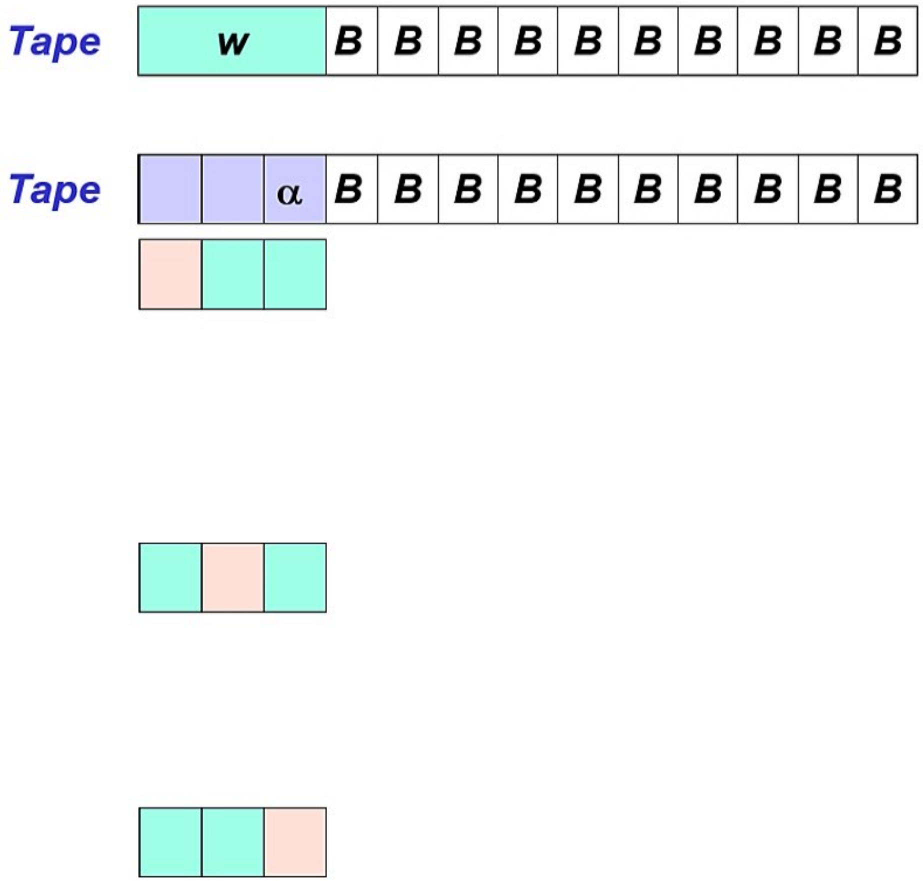
Tape	α	***B***	***B***	***B***	***B***	***B***	***B***	***B***	***B***	***B***	***B***

Construction of a Turing Machine from an Unrestricted Grammar

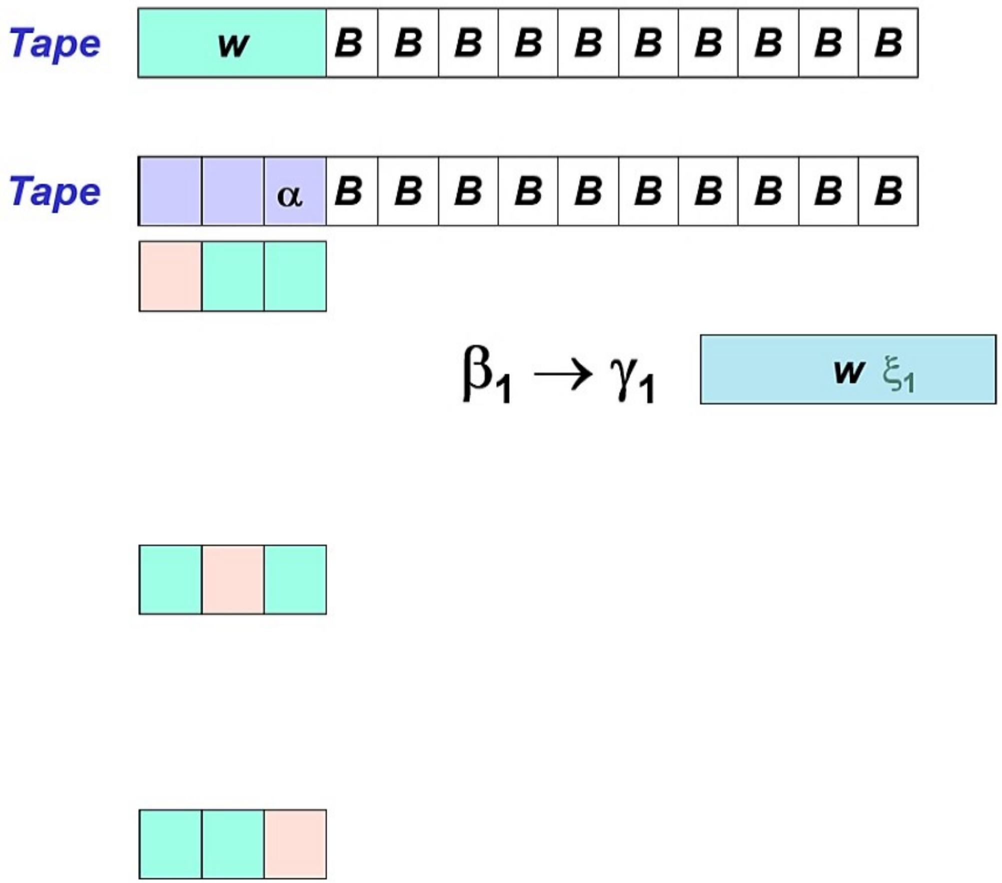
Tape	***w***	***B***	***B***	***B***	***B***	***B***	***B***	***B***	***B***	***B***	***B***

Tape	***w*** α	***B***	***B***	***B***	***B***	***B***	***B***	***B***	***B***	***B***	***B***

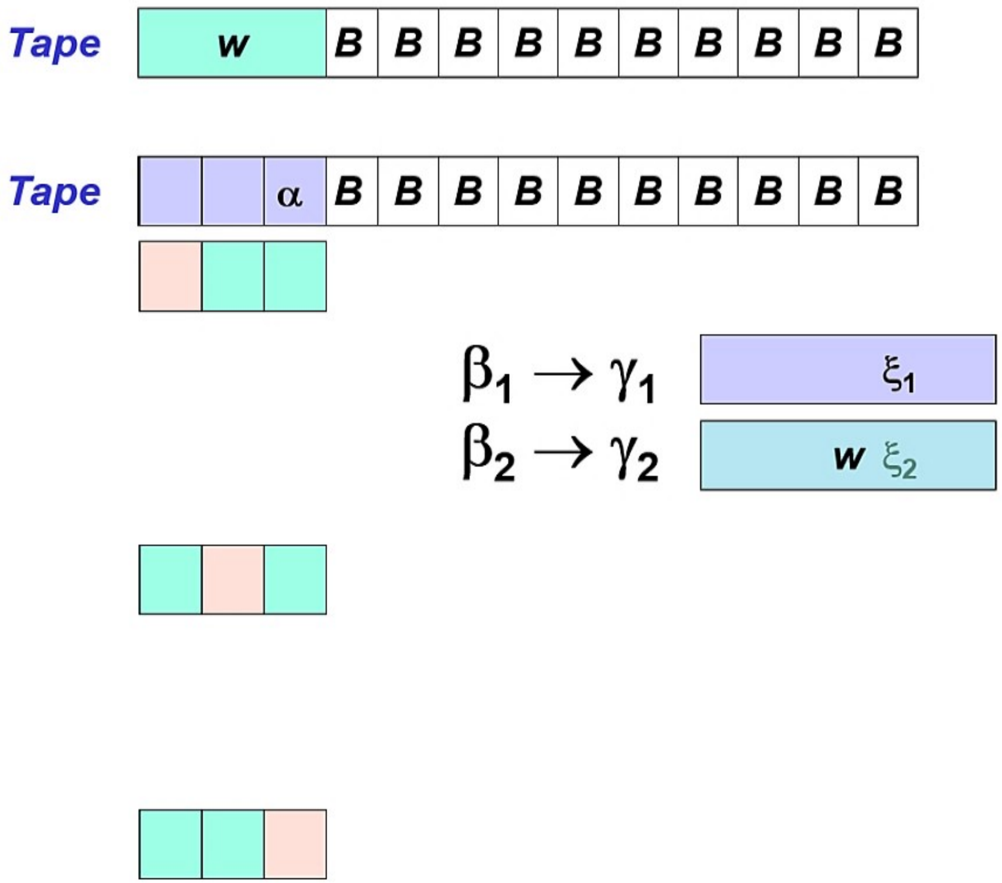
Construction of a Turing Machine from an Unrestricted Grammar



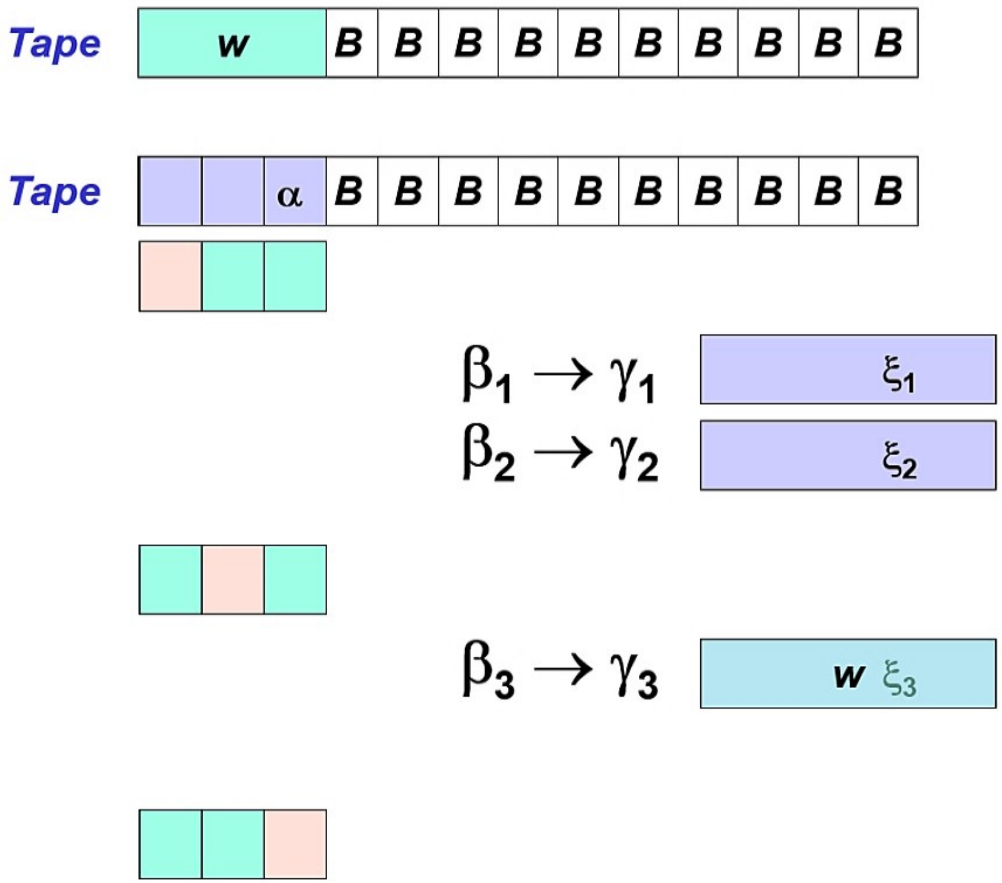
Construction of a Turing Machine from an Unrestricted Grammar



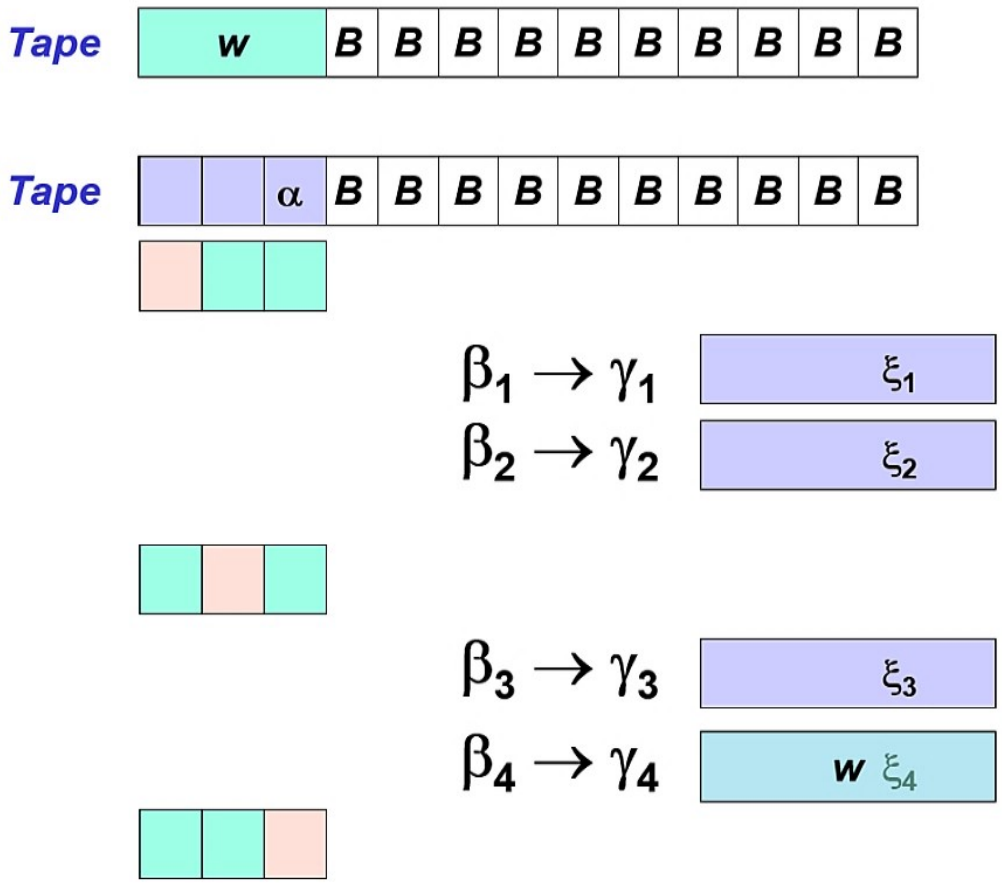
Construction of a Turing Machine from an Unrestricted Grammar



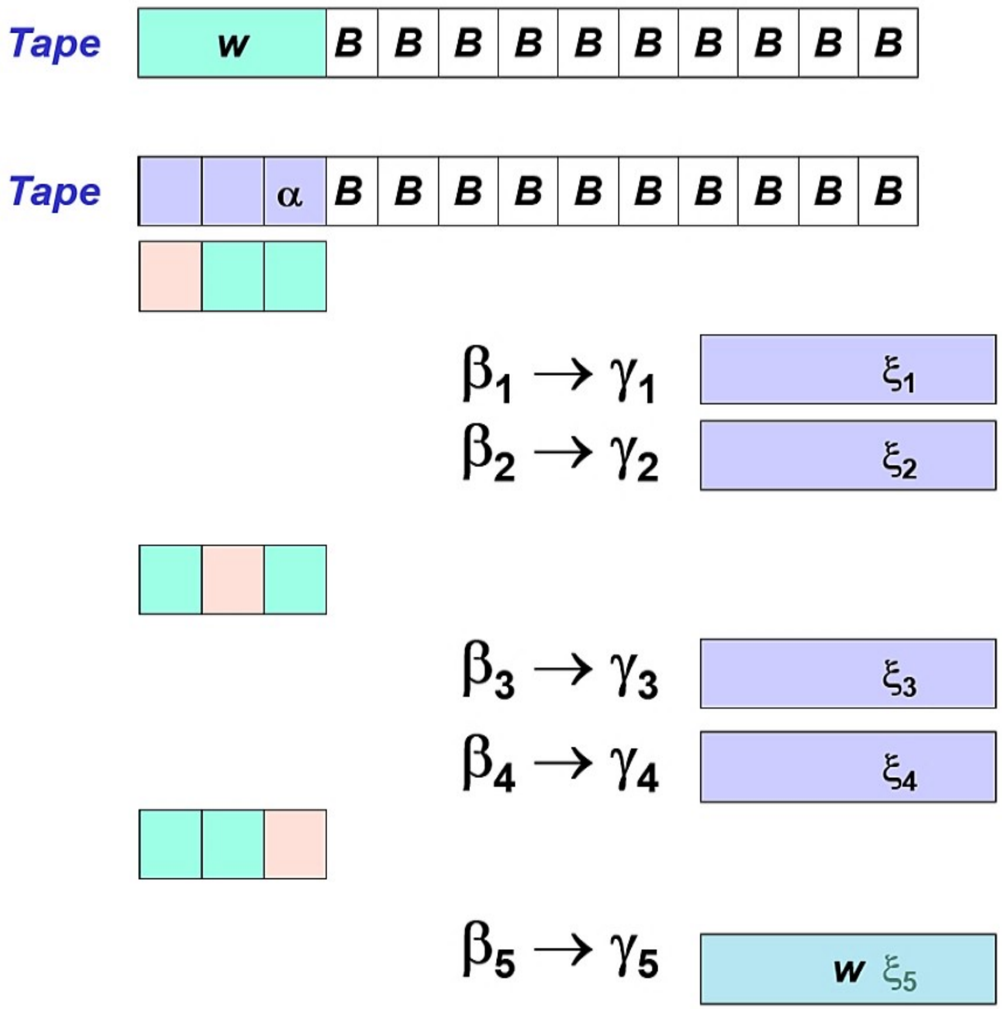
Construction of a Turing Machine from an Unrestricted Grammar



Construction of a Turing Machine from an Unrestricted Grammar



Construction of a Turing Machine from an Unrestricted Grammar



Lecture 14

Construction of a Grammar from a Turing Machine

Tape

a_1	a_2	--	a_n	B	B	B	B

State = q_0

Initial ID of a TM *M*: $q_0\ a_1\ a_2\text{---}\ a_n$

Derivative substring of a grammar: $q_0\ [a_1, a_1]\ [a_2, a_2]\ \text{---}\ [a_n, a_n]$

Construction of a Grammar from a Turing Machine

Tape	a_1	a_2	--	a_n	*B*	*B*	*B*	*B*	State = q_0
Tape	X	Y	--	Z	W	*B*	*B*	*B*	State = $q \in F$

Initial ID of a TM *M*: $q_0\ a_1\ a_2 \text{---}\ a_n$

Derivative substring of a grammar: $q_0\ [a_1, a_1]\ [a_2, a_2]\ \text{---}\ [a_n, a_n]$

Final ID of a TM *M*: $X\ Y \text{---}\ Z\ q\ W$

Derivative substring of a grammar: $[a_1, X]\ [a_2, Y]\ \text{---}\ [a_n, Z]\ q\ [\varepsilon, W]$

Construction of a Grammar from a Turing Machine

The string the grammar has to generate

Tape	a_1	a_2	--	a_n	B	B	B	B	State = q_0
Tape	X	Y	--	Z	W	B	B	B	State = $q \in F$

Initial ID of a TM *M*: $q_0\ a_1\ a_2 \text{---}\ a_n$

Derivative substring of a grammar: $q_0\ [a_1, a_1]\ [a_2, a_2]\ \text{---}\ [a_n, a_n]$

Final ID of a TM *M*: $X\ Y \text{---}\ Z\ q\ W$

Derivative substring of a grammar: $[a_1, X]\ [a_2, Y]\ \text{---}\ [a_n, Z]\ q\ [\varepsilon, W]$

Construction of a Grammar from a Turing Machine

Initial ID of a TM *M*: $q_0\ a_1\ a_2 \text{---}\ a_n$

Derivative substring of a grammar: $q_0\ [a_1, a_1]\ [a_2, a_2]\ \text{---}\ [a_n, a_n]$

Productions:

1) $A_1 \rightarrow q_0\ A_2$
2) $A_2 \rightarrow [a, a]\ A_2$

Derivative substring of a grammar: $q_0\ [a_1, a_1]\ [a_2, a_2]\ \text{---}\ [a_n, a_n]\ [\varepsilon, B]^m$

Productions:

3) $A_2 \rightarrow A_3$
4) $A_3 \rightarrow [\varepsilon, B]\ A_3$
5) $A_3 \rightarrow \varepsilon$

Construction of a Grammar from a Turing Machine

Transition of a TM *M* : $\delta(q, X) = (p, Y, R)$

Productions:

6) $q\,[\,a, X\,] \rightarrow [\,a, Y\,]\,p$

Transition of a TM *M* : $\delta(q, X) = (p, Y, L)$

Productions:

7) $[\,b, Z\,]\,q\,[\,a, X\,] \rightarrow p\,[\,b, Z\,]\,[\,a, Y\,]$

Construction of a Grammar from a Turing Machine

$q_0\ a_1\ a_2 \text{---}\ a_n$

$\succ X_1 \quad X_2 \quad \text{---}\ X_{r\text{-}1}\ q \quad X_r \quad \text{---}\ X_s$

$q_0\ [a_1, a_1]\ [a_2, a_2]\ \text{---}\ [a_n, a_n]\ [\varepsilon, B]^m$

$\Rightarrow [a_1, X_1]\ [a_2, X_2]\ \text{---}\ [a_n, X_n]\ \text{---}\ [\varepsilon, X_{r\text{-}1}]\ q\ [\varepsilon, X_r]\ \text{---}\ [\varepsilon, X_s]\ [\varepsilon, B]\ \text{---}\ [\varepsilon, B]$

Construction of a Grammar from a Turing Machine

$q_0 \boxed{a_1\ a_2 \text{---}\ a_n}$

$\succ X_1 \quad X_2 \quad \text{---}\ X_{r-1}\ q \quad X_r \quad \text{---}\ X_s$

$q_0 [\boxed{a_1,}\ a_1]\ [\boxed{a_2,}\ a_2]\ \text{---}\ [\boxed{a_n,}\ a_n]\ [\varepsilon, B]^m$

$\Rightarrow [\boxed{a_1,}\ X_1]\ [\boxed{a_2,}\ X_2]\ \text{---}\ [\boxed{a_n,}\ X_n]\ \text{---}\ [\varepsilon, X_{r-1}]\ q\ [\varepsilon, X_r]\ \text{---}\ [\varepsilon, X_s]\ [\varepsilon, B]\ \text{---}\ [\varepsilon, B]$

Construction of a Grammar from a Turing Machine

$q_0 \boxed{a_1\ a_2 \text{---}\ a_n}$

$\succ \boxed{X_1 \quad X_2 \quad \text{---}\ X_{r-1}\ q \quad X_r \quad \text{---}\ X_s}$

$q_0 [\boxed{a_1,}\ a_1]\ [\boxed{a_2,}\ a_2]\ \text{---}\ [\boxed{a_n,}\ a_n]\ [\varepsilon, B]^m$

$\Rightarrow [\boxed{a_1,}\boxed{X_1}]\ [\boxed{a_2,}\boxed{X_2}]\ \text{---}\ [\boxed{a_n,}\boxed{X_n}]\ \text{---}\ [\varepsilon,\boxed{X_{r-1}}]\ q\ [\varepsilon,\boxed{X_r}]\ \text{---}\ [\varepsilon,\boxed{X_s}]\ [\varepsilon, B]\ \text{---}\ [\varepsilon, B]$

Construction of a Grammar from a Turing Machine

For all accepting states $q \in F$

For all input symbols $a \in \Sigma \cup \{ \varepsilon \}$

For all tape symbols $X \in \Gamma$

Productions:

8)	$[a, X]\ q \to q\ a\ q$
9)	$q\ [a, X] \to q\ a\ q$
10)	$q \to \varepsilon$

Construction of a Grammar from a Turing Machine

For all accepting states $q \in F$

For all input symbols $a \in \Sigma \cup \{ \varepsilon \}$

For all tape symbols $X \in \Gamma$

Productions:

8)	$[a, X]\, q \to q\, a\, q$
9)	$q\, [a, X] \to q\, a\, q$
10)	$q \to \varepsilon$

$$[a_1, X_1]\,[a_2, X_2] \text{ --- } [a_{t-1}, X_{t-1}]\; q\; [a_t, X_t]\,[a_{t+1}, X_{t+1}] \text{ --- } [a_n, X_n]\,[\varepsilon, X_n] \text{ --- } [\varepsilon, X_{n+m}]$$

Construction of a Grammar from a Turing Machine

For all accepting states $q \in F$

For all input symbols $a \in \Sigma \cup \{ \varepsilon \}$

For all tape symbols $X \in \Gamma$

Productions:

8)	$[a, X]\ q \rightarrow q\ a\ q$
9)	$q\ [a, X] \rightarrow q\ a\ q$
10)	$q \rightarrow \varepsilon$

$$[a_1, X_1]\ [a_2, X_2] \text{ --- } [a_{t-1}, X_{t-1}]\ q \quad a_t \quad q\ [a_{t+1}, X_{t+1}] \text{ --- } [a_n, X_n]\ [\varepsilon, X_n] \text{ --- } [\varepsilon, X_{n+m}]$$

Construction of a Grammar from a Turing Machine

For all accepting states $q \in F$

For all input symbols $a \in \Sigma \cup \{ \varepsilon \}$

For all tape symbols $X \in \Gamma$

Productions:

8)	$[a, X]\ q \rightarrow q\ a\ q$
9)	$q\ [a, X] \rightarrow q\ a\ q$
10)	$q \rightarrow \varepsilon$

$$[a_1, X_1]\ [a_2, X_2] \text{ --- } q \quad a_{t-1} \quad q \quad a_t \quad q \quad a_{t+1} \quad q \text{ --- } [a_n, X_n]\ [\varepsilon, X_n] \text{ --- } [\varepsilon, X_{n+m}]$$

Construction of a Grammar from a Turing Machine

For all accepting states $q \in F$

For all input symbols $a \in \Sigma \cup \{ \varepsilon \}$

For all tape symbols $X \in \Gamma$

Productions:

8)	$[a, X]\ q \rightarrow q\ a\ q$
9)	$q\ [a, X] \rightarrow q\ a\ q$
10)	$q \rightarrow \varepsilon$

$$q\ a_1\ q\ a_2\ \text{---}\ q\ a_{t-1}\ q\ a_t\ q\ a_{t+1}\ q\ \text{---}\ q\ a_n\ q\ q\ q\ q$$

Construction of a Grammar from a Turing Machine

For all accepting states $q \in F$

For all input symbols $a \in \Sigma \cup \{ \varepsilon \}$

For all tape symbols $X \in \Gamma$

Productions:

8)	$[a, X]\, q \rightarrow q\, a\, q$
9)	$q\, [a, X] \rightarrow q\, a\, q$
10)	$q \rightarrow \varepsilon$

a_1 a_2 --- a_{t-1} a_t a_{t+1} --- a_n

Construction of a Grammar from a Turing Machine

For all accepting states $q \in F$

For all input symbols $a \in \Sigma \cup \{ \varepsilon \}$

For all tape symbols $X \in \Gamma$

Productions:

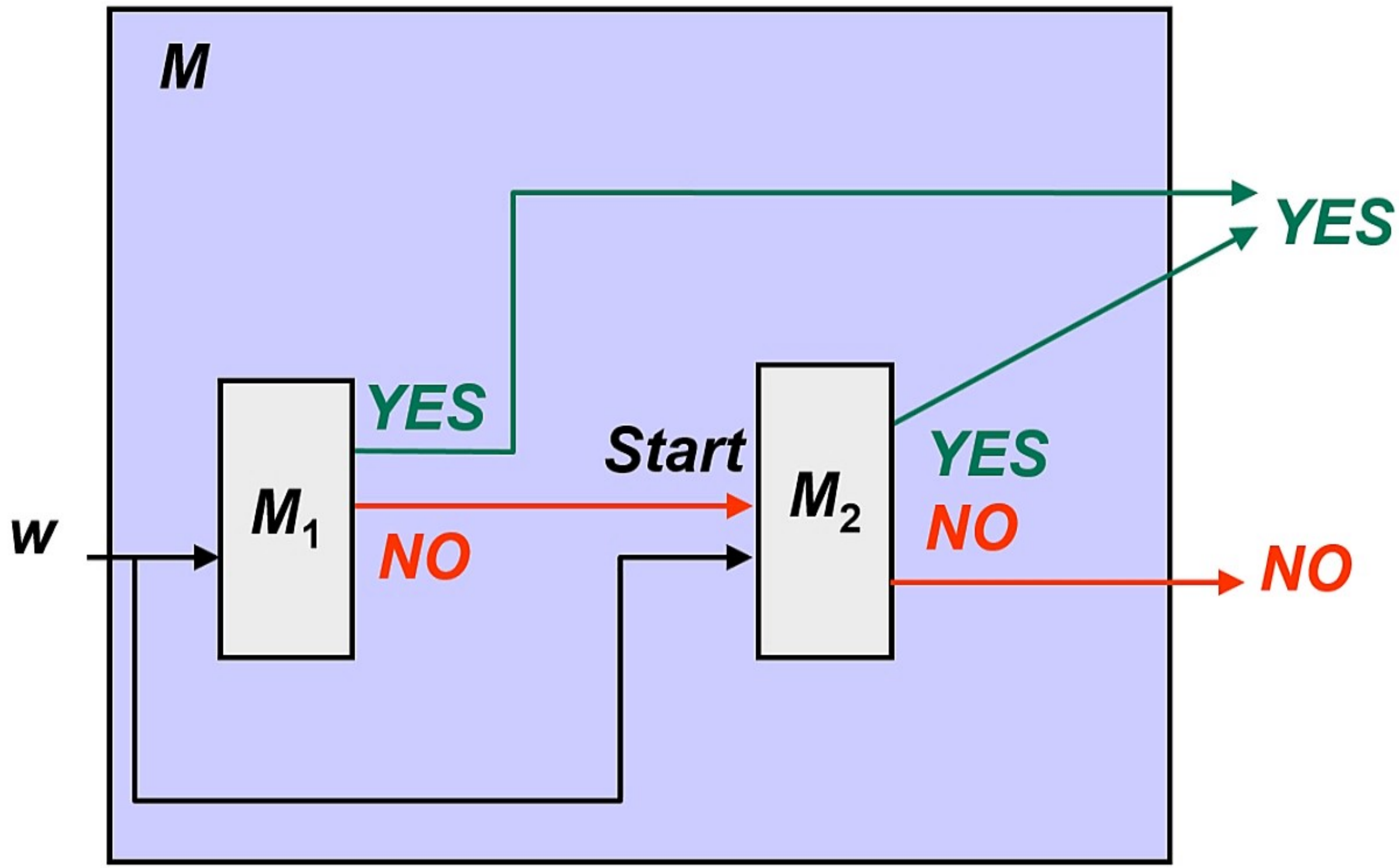
8)	$[a, X]\ q \rightarrow q\ a\ q$
9)	$q\ [a, X] \rightarrow q\ a\ q$
10)	$q \rightarrow \varepsilon$

a_1 a_2 --- a_{t-1} a_t a_{t+1} --- a_n

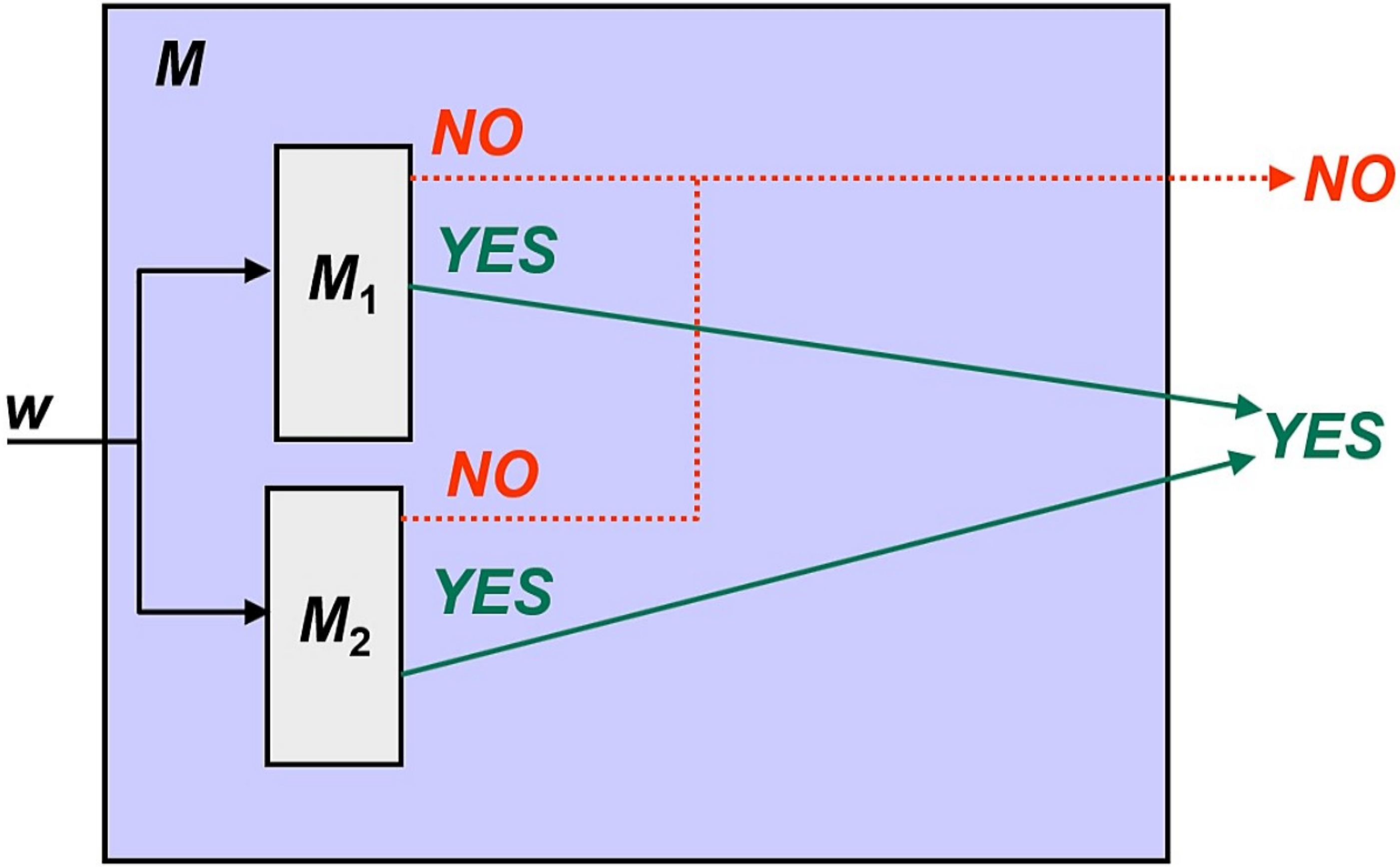
$\Rightarrow\ a_1\ a_2 \text{---}\ a_n$

Lecture 14

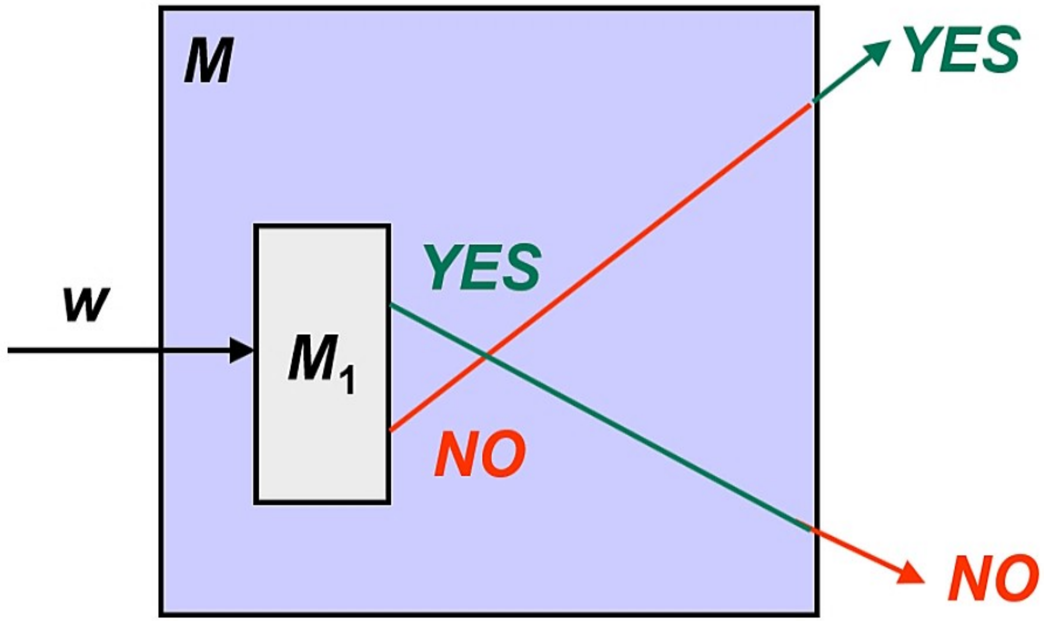
Union of Recursive Languages



Union of Recursively Enumerable Languages

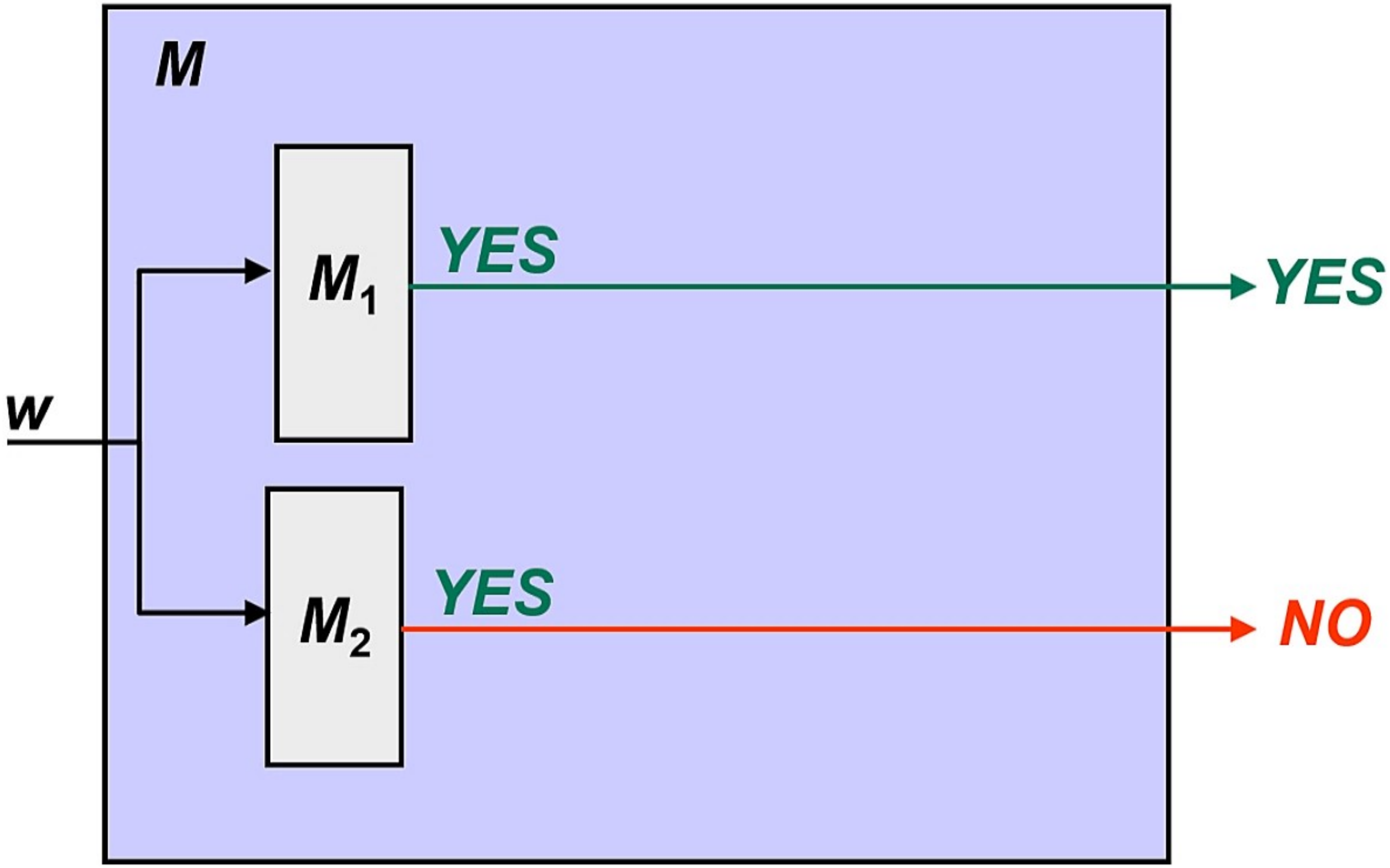


Complement of a Recursive Language



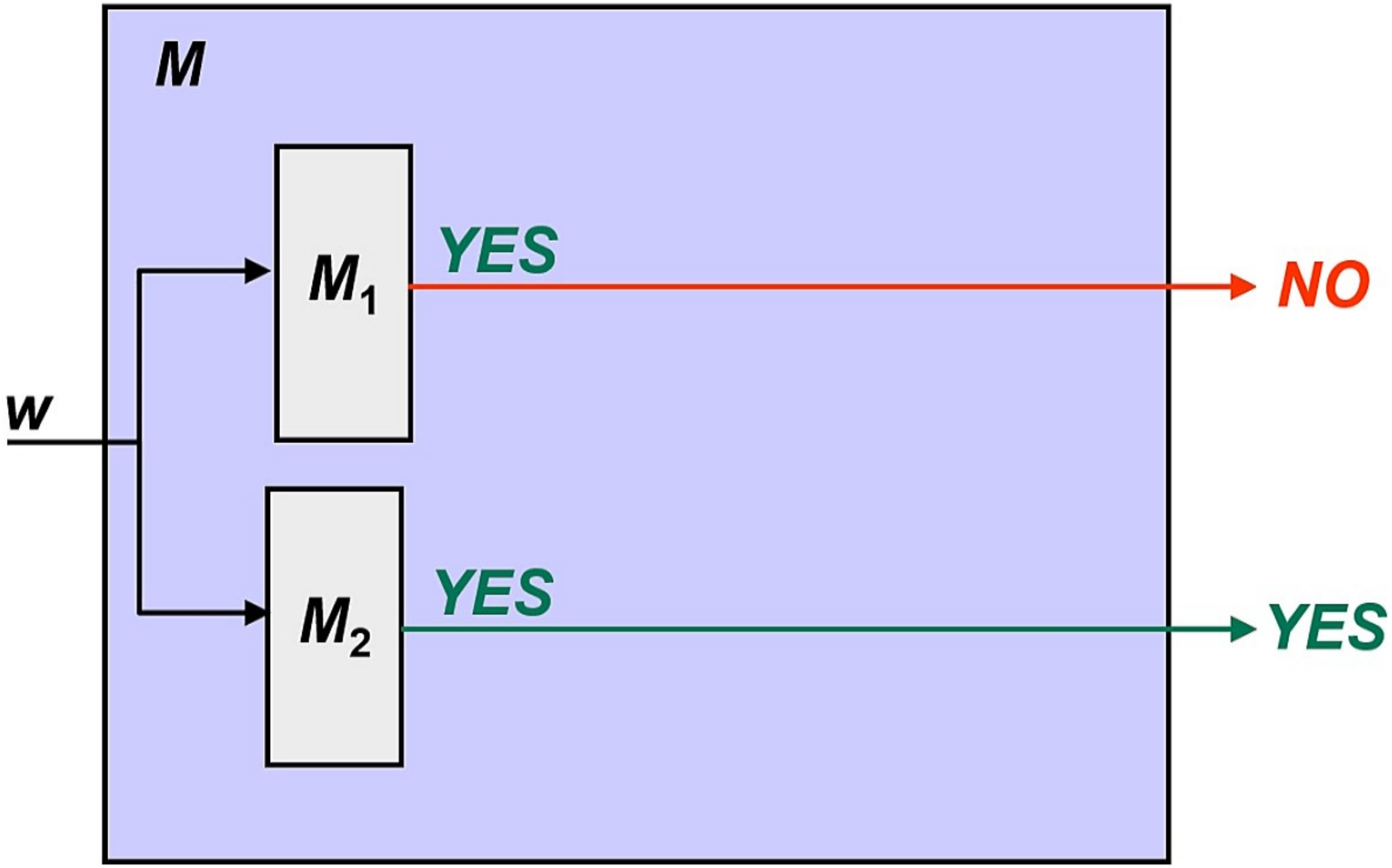
Complement of a Recursive Language

Language L_1 - recursively enumerable
Language L_2, which is a complement of L_1 - recursively enumerable



Complement of a Recursive Language

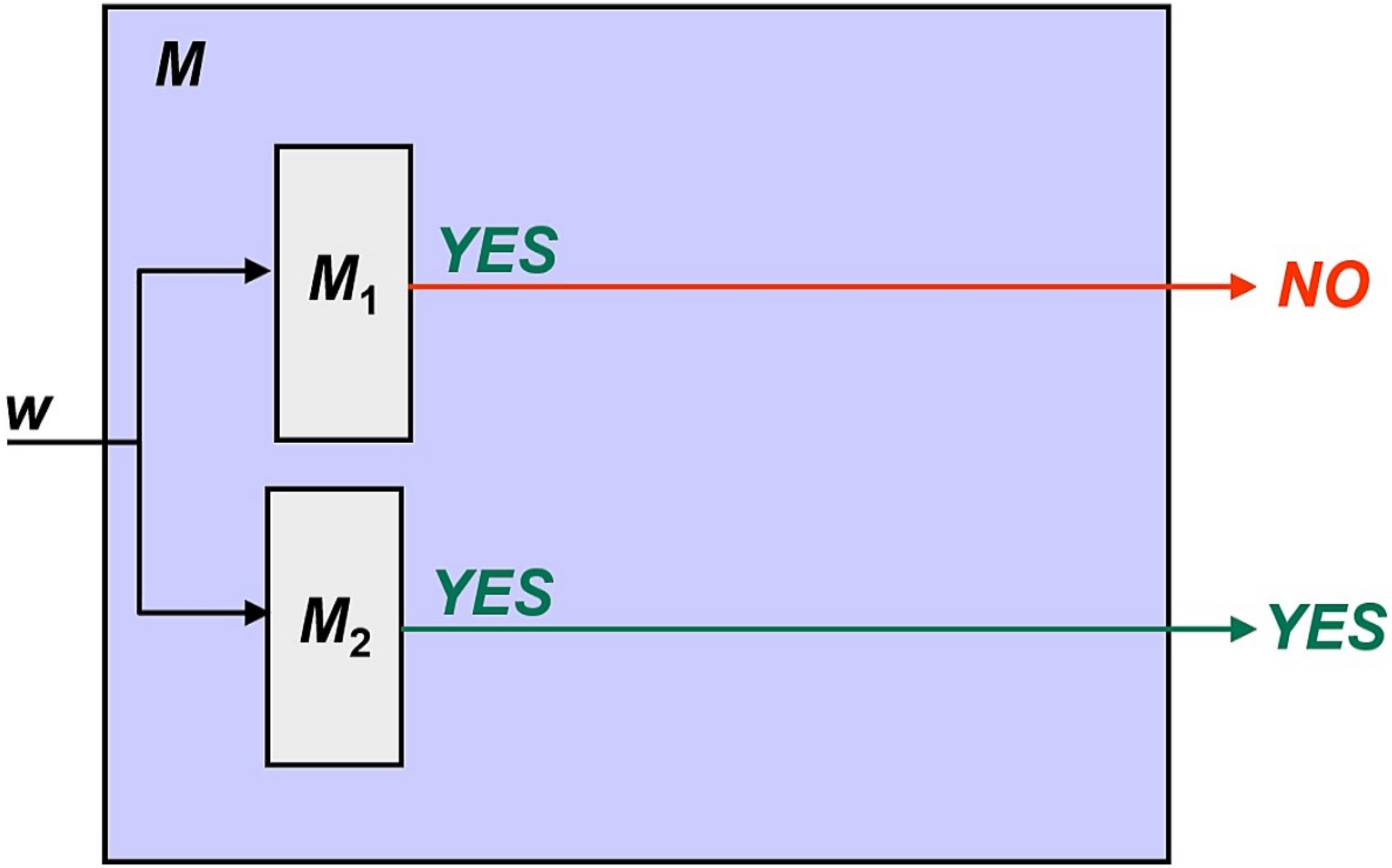
Language L_1 - recursively enumerable
Language L_2, which is a complement of L_1 - recursively enumerable



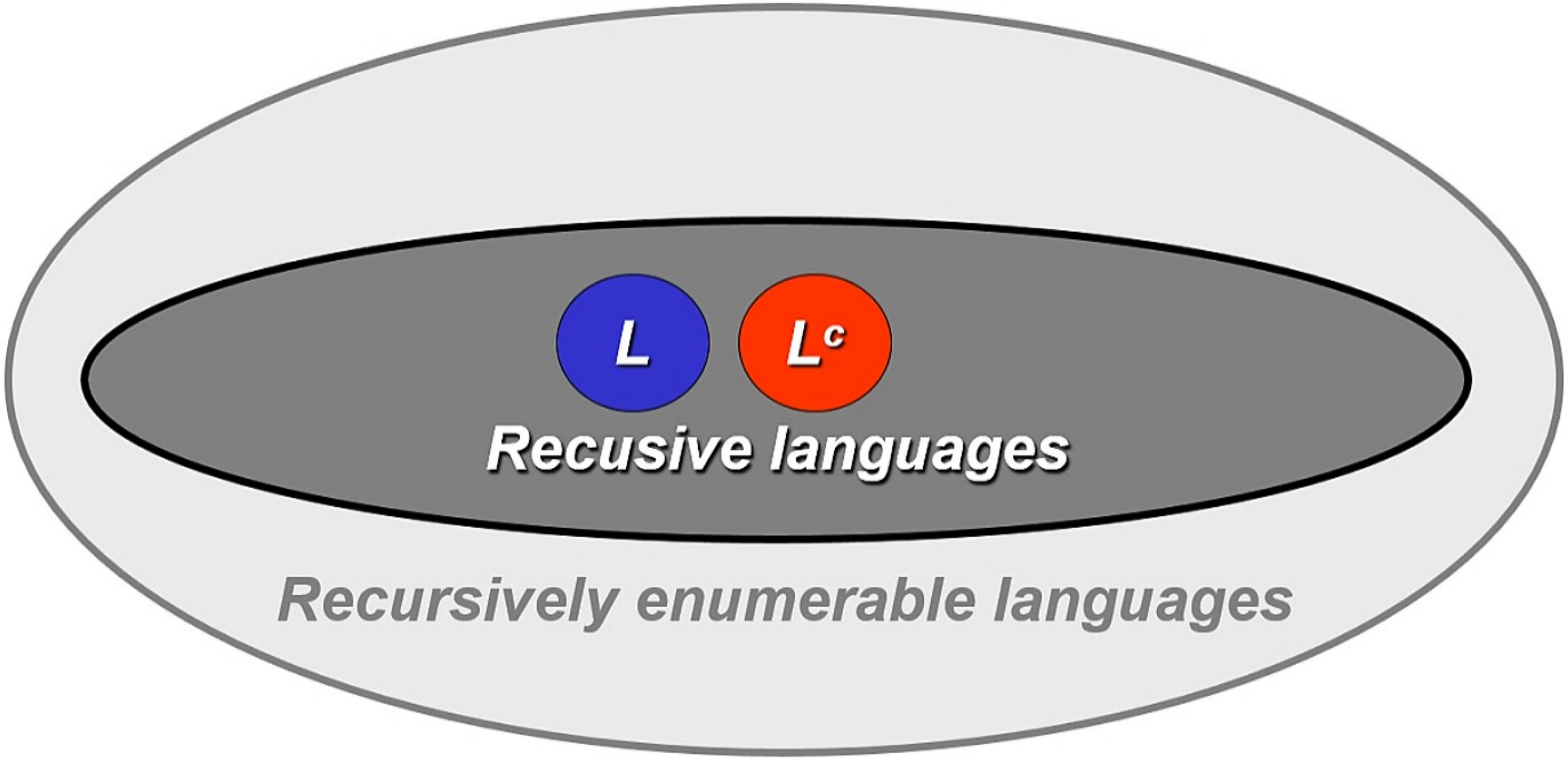
Complement of a Recursive Language

Language L_1 - recursively enumerable
Language L_2, which is a complement of L_1 - recursively enumerable

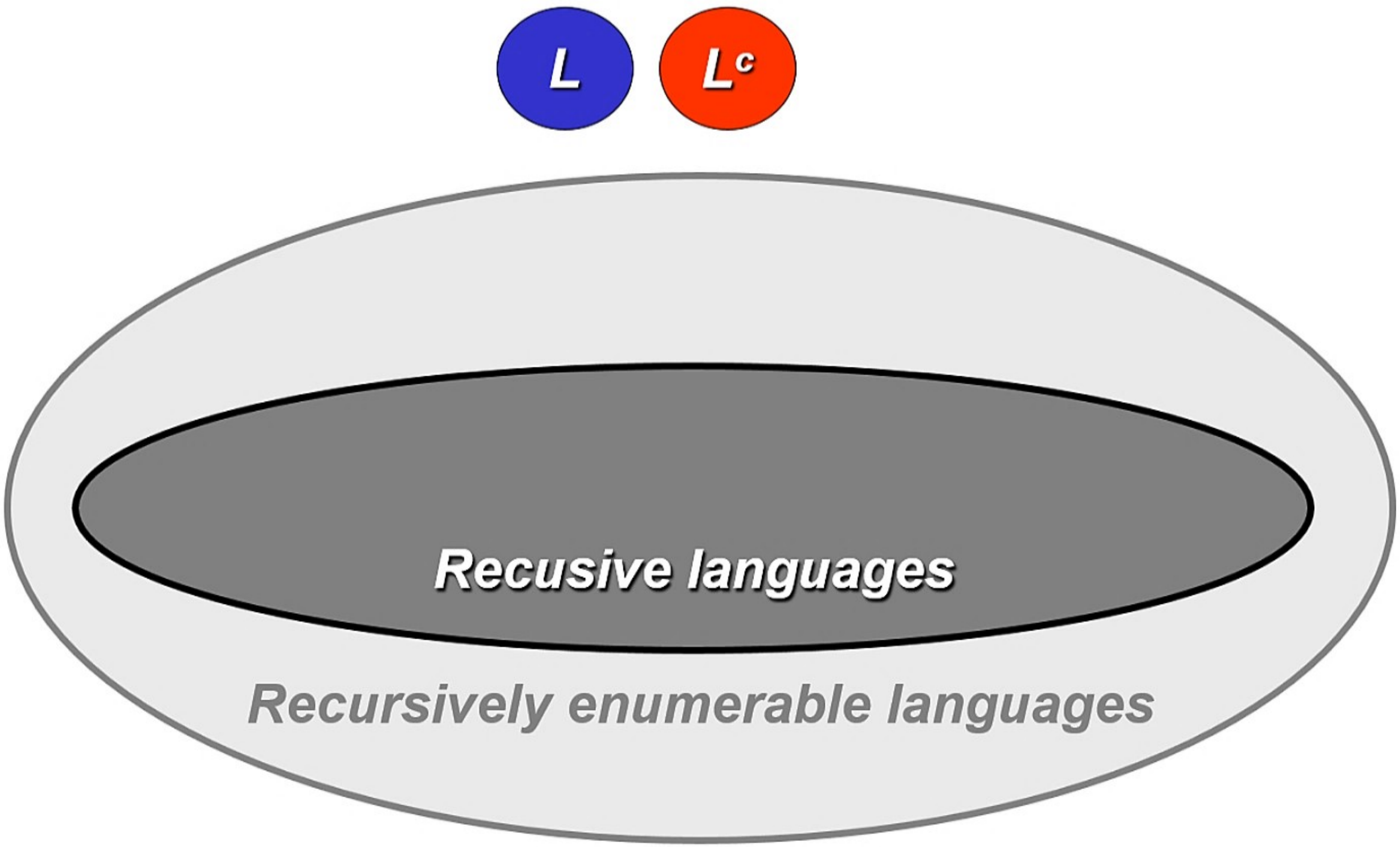
then both languages are RECURSIVE



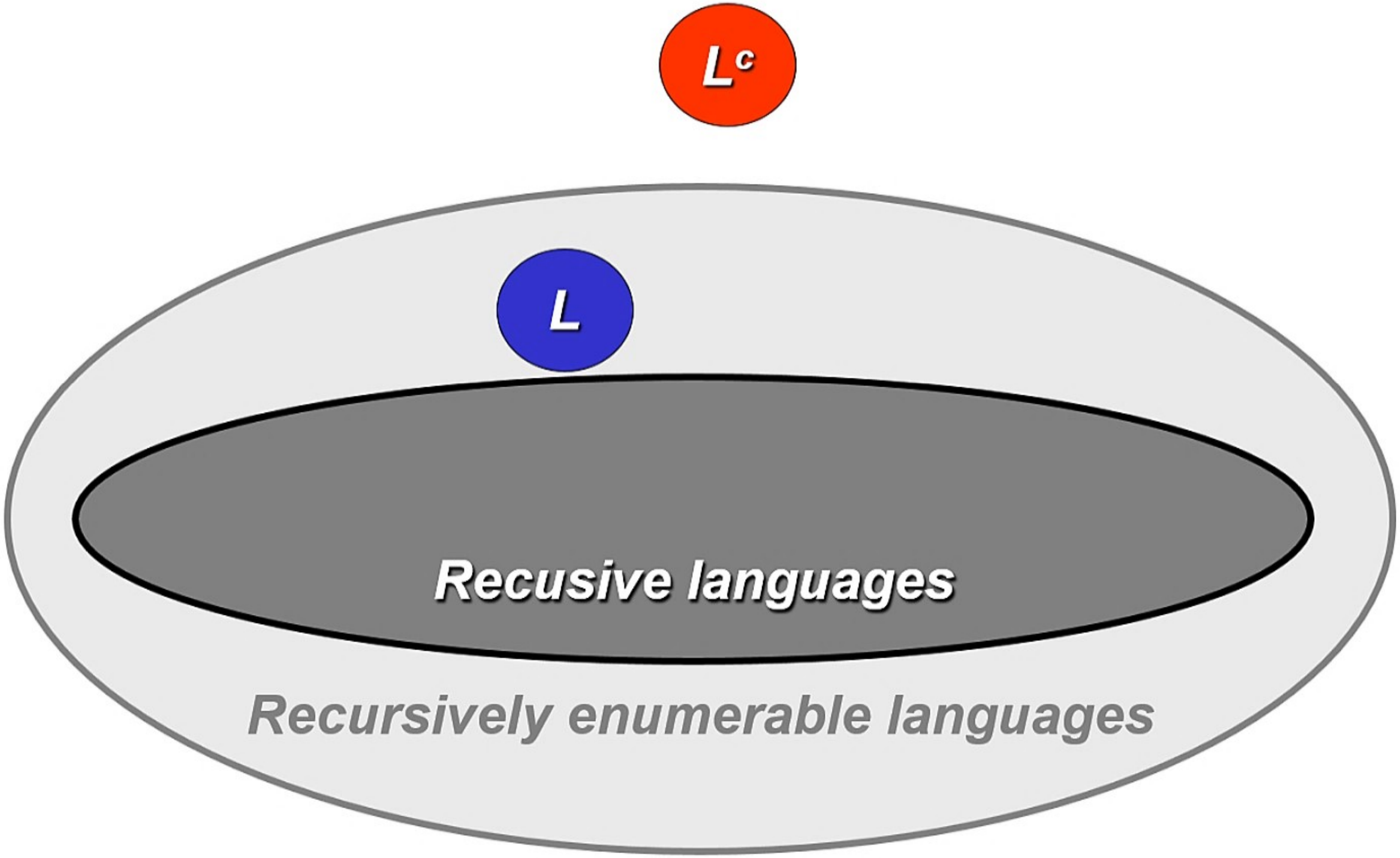
Complement


L
Lᶜ
Recusive languages
Recursively enumerable languages

Complement


L
Lc
Recusive languages
Recursively enumerable languages

Complement



Lc
L
Recusive languages
Recursively enumerable languages

Lecture 14

Computability

- **A problem is computable**
  - if there exists an automaton that solves the problem step by step
  - no limitations on
    - —the number of steps
    - —the size of the tape (memory)
    - —we do not require that the process ever halts

Computability

- **Church-Turing hypothesis**
  - computable functions – partial (general) recursive functions
  - we can build a TM to compute them
  - TM computes partial recursive functions step by step using a given transition function
  - TM does not have to halt for all input strings
    - computability of partial recursive functions is based on the broadest definition

Simulation of a Digital Computer by a Turing Machine

Memory | # $0 * v_0$ # $1 * v_1$ # $10 *$ v_2 # - - - # $k * v_k$

Arithmetic registers | # R_0 # R_1 # R_2 # - - - # R_l | B | B | B

Program counter | PC | B | B | B | B | B | B | B | B | B | B | B

Memory address register | MAR | B | B | B | B | B | B | B | B | B

Mikrocode | # $n_0 * k_0$ # $n_1 *$ k_1 # $n_2 * k_2$ # - - - # $n_m * k_m$

Simulation of a Digital Computer by a Turing Machine

Memory | # $0 * v_0$ # $1 * v_1$ # $10 *$ v_2 # - - - # $k * v_k$

Arithmetic registers | # R_0 # R_1 # R_2 # - - - # R_l | B | B | B

Program counter | 10 | B | B | B | B | B | B | B | B | B | B | B

Memory address register | MAR | B | B | B | B | B | B | B | B | B

Mikrocode | # $n_0 * k_0$ # $n_1 *$ k_1 # $n_2 * k_2$ # - - - # $n_m * k_m$

Simulation of a Digital Computer by a Turing Machine

Memory # $0 * v_0$ # $1 * v_1$ # $10 * v_2$ # - - - # $k * v_k$

Arithmetic registers # R_0 # R_1 # R_2 # - - - # R_l | *B* | *B* | *B*

Program counter | 10 | *B* | *B* | *B* | *B* | *B* | *B* | *B* | *B* | *B* | *B* | *B*

Memory address register | 10 | *B* | *B* | *B* | *B* | *B* | *B* | *B* | *B* | *B*

Mikrocode # $n_0 * k_0$ # $n_1 * k_1$ # $n_2 * k_2$ # - - - # $n_m * k_m$

Simulation of a Digital Computer by a Turing Machine

Memory | # $0 * v_0$ # $1 * v_1$ # $10 * v_2$ # - - - # $k * v_k$

Arithmetic registers | # R_0 # R_1 # R_2 # - - - # R_l | B | B | B

Program counter

11	B	B	B	B	B	B	B	B	B	B	B

Memory address register

10	B	B	B	B	B	B	B	B	B

Mikrocode | # $n_0 * k_0$ # $n_1 * k_1$ # $n_2 * k_2$ # - - - # $n_m * k_m$

Simulation of a Digital Computer by a Turing Machine

Memory: # 10 * v_0 # 1 * v_1 # 10 * v_2 # - - - # k * v_k

Arithmetic registers: # R_0 # R_1 # R_2 # - - - # R_l | B | B | B

Program counter: 11 | B | B | B | B | B | B | B | B | B | B | B

Memory address register: 10 | B | B | B | B | B | B | B | B | B

Mikrocode: # n_0 * k_0 # n_1 * k_1 # n_2 * k_2 # - - - # n_m * k_m

Simulation of a Digital Computer by a Turing Machine

Memory | # 0 * v_0 # 10 v_1 # 10 * v_2 # - - - # k * v_k

Arithmetic registers | # R_0 # R_1 # R_2 # - - - # R_l | B | B | B

Program counter | 11 | B | B | B | B | B | B | B | B | B | B | B

Memory address register | 10 | B | B | B | B | B | B | B | B | B

Mikrocode | # n_0 * k_0 # n_1 * k_1 # n_2 * k_2 # - - - # n_m * k_m

Simulation of a Digital Computer by a Turing Machine

Memory # $0 * v_0$ # $1 * v_1$ # $10 *$ v_2 # - - - # $k * v_k$

Arithmetic registers # R_0 # R_1 # R_2 # - - - # R_l | B | B | B

Program counter | 11 | B | B | B | B | B | B | B | B | B | B | B

Memory address register | 10 | B | B | B | B | B | B | B | B | B

Mikrocode # $n_0 * k_0$ # $n_1 *$ k_1 # $n_2 * k_2$ # - - - # $n_m * k_m$

Simulation of a Digital Computer by a Turing Machine

Memory	# 0 * v_0 # 1 * v_1 # 10 *	n_1 p	# - - - # k * v_k

Arithmetic registers	# R_0 # R_1 # R_2 # - - - # R_l	*B*	*B*	*B*

Program counter	11	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*

Memory address register	10	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*

Mikrocode	# n_0 * k_0 # n_1 * k_1 # n_2 * k_2 # - - - # n_m * k_m

Simulation of a Digital Computer by a Turing Machine

Memory | # $0 * v_0$ # $1 * v_1$ # $10 *$ n_1 p # - - - # $k * v_k$

Arithmetic registers | # R_0 # R_1 # R_2 # - - - # R_l | B | B | B

Program counter | 11 | B | B | B | B | B | B | B | B | B | B | B

Memory address register | 10 | B | B | B | B | B | B | B | B | B

Mikrocode | # $n_0 * k_0$ # $n_1 * k_1$ # $n_2 * k_2$ # - - - # $n_m * k_m$

Simulation of a Digital Computer by a Turing Machine

Memory # $0 * v_0$ # $1 * v_1$ # $10 *$ n_1 p # - - - # $k * v_k$

Arithmetic registers # R_0 # R_1 # R_2 # - - - # R_l | *B* | *B* | *B*

Program counter 11 | *B* | *B* | *B* | *B* | *B* | *B* | *B* | *B* | *B* | *B* | *B*

Memory address register 10 | *B* | *B* | *B* | *B* | *B* | *B* | *B* | *B* | *B*

Mikrocode # $n_0 * k_0$ # $n_1 *$ k_1 # $n_2 * k_2$ # - - - # $n_m * k_m$

Simulation of a Digital Computer by a Turing Machine

Memory	# $0 * v_0$ # $1 * v_1$ # $10 *$	n_1 p	# - - - # $k * v_k$

Arithmetic registers	# R_0 # R_1 # R_2 # - - - # R_l	*B*	*B*	*B*

Program counter	11	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*

Memory address register	10	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*

Mikrocode	# $n_0 * k_0$ # $n_1 *$	*ADD*	# $n_2 * k_2$ # - - - # $n_m * k_m$

Simulation of a Digital Computer by a Turing Machine

Memory

# 0 * v_0 # 1 * v_1 # 10 *	n_1	k, R_2	# - - - # k * v_k

Arithmetic registers

# R_0 # R_1 # R_2 # - - - # R_l	*B*	*B*	*B*

Program counter

11	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*

Memory address register

10	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*

Mikrocode

# n_0 * k_0 # n_1 *	*ADD*	# n_2 * k_2 # - - - # n_m * k_m

Simulation of a Digital Computer by a Turing Machine

Memory	# $0 * v_0$ # $1 * v_1$ # $10 *$	n_1	k, R_2	# - - - # $k * v_k$

Arithmetic registers	# R_0 # R_1 # R_2 # - - - # R_l	***B***	***B***	***B***

Program counter	11	***B***	***B***	***B***	***B***	***B***	***B***	***B***	***B***	***B***	***B***	***B***

Memory address register	k	***B***	***B***	***B***	***B***	***B***	***B***	***B***	***B***	***B***

Mikrocode	# $n_0 * k_0$ # $n_1 *$	***ADD***	# $n_2 * k_2$ # - - - # $n_m * k_m$

Simulation of a Digital Computer by a Turing Machine

Memory

#	0k*	v_0 # 1 * v_1 # 10 *	n_1	k, R_2	# - - - # k * v_k

Arithmetic registers

# R_0 # R_1 # R_2 # - - - # R_l	*B*	*B*	*B*

Program counter

11	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*

Memory address register

k	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*

Mikrocode

# n_0 * k_0 # n_1 *	*ADD*	# n_2 * k_2 # - - - # n_m * k_m

Simulation of a Digital Computer by a Turing Machine

Memory | # 0 * v_0 # 1k* v_1 # 10 * | n_1 | k, R_2 | # - - - # k * v_k

Arithmetic registers | # R_0 # R_1 # R_2 # - - - # R_l | B | B | B

Program counter | 11 | B | B | B | B | B | B | B | B | B | B | B

Memory address register | k | B | B | B | B | B | B | B | B | B

Mikrocode | # n_0 * k_0 # n_1 * | ADD | # n_2 * k_2 # - - - # n_m * k_m

Simulation of a Digital Computer by a Turing Machine

Memory	# 0 * v_0 # 1 * v_1 # k * n_1	k, R_2	# - - - # k * v_k

Arithmetic registers	# R_0 # R_1 # R_2 # - - - # R_l	B	B	B

Program counter	11	B	B	B	B	B	B	B	B	B	B	B

Memory address register	k	B	B	B	B	B	B	B	B	B

Mikrocode	# n_0 * k_0 # n_1 *	ADD	# n_2 * k_2 # - - - # n_m * k_m

Simulation of a Digital Computer by a Turing Machine

Memory	# $0 * v_0$ # $1 * v_1$ # $10 *$ n_1 k, R_2 # - - - # $k * v_k$
Arithmetic registers	# R_0 # R_1 # R_2 # - - - # R_l B B B
Program counter	11 B B B B B B B B B B B
Memory address register	k B B B B B B B B B
Mikrocode	# $n_0 * k_0$ # $n_1 *$ ADD # $n_2 * k_2$ # - - - # $n_m * k_m$

Simulation of a Digital Computer by a Turing Machine

Memory

# 0 * v_0 # 1 * v_1 # 10 *	n_1	k, R_2	# - - - # k *	3

Arithmetic registers

# R_0 # R_1 # R_2 # - - - # R_l	*B*	*B*	*B*

Program counter

11	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*

Memory address register

k	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*

Mikrocode

# n_0 * k_0 # n_1 *	*ADD*	# n_2 * k_2 # - - - # n_m * k_m

Simulation of a Digital Computer by a Turing Machine

Memory	# 0 * v_0 # 1 * v_1 # 10 * n_1 k, R_2 # - - - # k * 3
Arithmetic registers	# R_0 # R_1 # R_2 # - - - # R_l B B B
Program counter	11 B B B B B B B B B B B
Memory address register	k B B B B B B B B B
Mikrocode	# n_0 * k_0 # n_1 * ADD # n_2 * k_2 # - - - # n_m * k_m

Simulation of a Digital Computer by a Turing Machine

Memory

# 0 * v_0 # 1 * v_1 # 10 *	n_1	k, R_2	# - - - # k *	3

Arithmetic registers

# R_0	#	R_1 # R_2 # - - - # R_l	B	B	B

Program counter

11	B	B	B	B	B	B	B	B	B	B	B

Memory address register

k	B	B	B	B	B	B	B	B	B

Mikrocode

# n_0 * k_0 # n_1 *	ADD	# n_2 * k_2 # - - - # n_m * k_m

Simulation of a Digital Computer by a Turing Machine

Memory	# 0 * v_0 # 1 * v_1 # 10 * | n_1 | k, R_2 | # - - - # k * | 3
Arithmetic registers	# R_0 # R_1 | # | R_2 # - - - # R_l | B | B | B
Program counter	11 | B | B | B | B | B | B | B | B | B | B | B
Memory address register	k | B | B | B | B | B | B | B | B | B
Mikrocode	# n_0 * k_0 # n_1 * | ADD | # n_2 * k_2 # - - - # n_m * k_m

Simulation of a Digital Computer by a Turing Machine

Memory	# $0 * v_0$ # $1 * v_1$ # $10 *$ | n_1 | k, R_2 | # - - - # $k *$ | 3
Arithmetic registers	# R_0 # R_1 # | $R_2 = R_2 + 3$ | # - - - # R_l | B | B | B
Program counter	11 | B | B | B | B | B | B | B | B | B | B | B
Memory address register	k | B | B | B | B | B | B | B | B | B
Mikrocode	# $n_0 * k_0$ # $n_1 *$ | ADD | # $n_2 * k_2$ # - - - # $n_m * k_m$

Simulation of a Digital Computer by a Turing Machine

Memory

# 0 * v_0 # 1 * v_1 # 10 *	n_1	k, R_2	# - - - # k *	3

Arithmetic registers

# R_0 # R_1 #	$R_2 = R_2 + 3$	# - - - # R_l	B	B	B

Program counter

11	B	B	B	B	B	B	B	B	B	B	B

Memory address register

11	B	B	B	B	B	B	B	B	B

Mikrocode

# n_0 * k_0 # n_1 *	ADD	# n_2 * k_2 # - - - # n_m * k_m

Uncomputability of Diagonal Language

- **Coding of a TM *M***
  - **Coding of symbols**

	Symbol	*Code*
Tape symbols	0 1 *B*	0 00 000
Head shift direction	*L* *R*	0 00
States	q_1 q_2 - - - q_n	0 00 - - - 0^n

Uncomputability of Diagonal Language

- Coding of a TM *M*
  - **Coding of a transition function**

A transition:

$$\delta(q_i, X_j) = (q_k, X_l, D_m)$$

is represented as a binary string:

$$0^i 1 0^j 1 0^k 1 0^l 1 0^m$$

Uncomputability of Diagonal Language

- **Coding of a TM *M***

$$111 \;\; code_1 \;\; 11 \;\; code_2 \;\; 11 \; - - - \; 11 \;\; code_r \;\; 111$$

$\delta(q_1, 1)=(q_3, 0, R)$, $\delta(q_3, 0)=(q_1, 1, R)$, $\delta(q_3, 1)=(q_2, 0, R)$, $\delta(q_3, B)=(q_3, 1, L)$

111 0100100010100 11 0001010100100 11 00010010010100 11 0001000100010010 111

Uncomputability of Diagonal Language

- **Construction of the diagonal language**

$j \rightarrow$ (columns), $i \downarrow$ (rows)

	1	2	-	x	x+1	x+2	x+3	-
1	0	0	-	0	0	0	0	-
2	0	0	-	0	0	0	0	-
-	-	-	-	-	-	-	-	-
x	0	0	-	0	0	0	1	-
x+1	0	0	-	1	1	1	0	-
x+2	0	0	-	1	0	0	0	-
x+3	0	0	-	0	0	0	1	-
-	-	-	-	-	-	-	-	-

Uncomputability of a Diagonal Language

- **TM M_d that accepts the diagonal language $L_d = L(M_d)$**

$j \rightarrow$

$i \downarrow$		1	2	-	d	-
	1	0	0	-		-
	2	0	0	-		-
	-	-	-	-		-
	d				?	
	-	-	-	-		-

Uncomputability of a Diagonal Language

- **TM M_d that accepts the diagonal language $L_d = L(M_d)$**

$j \rightarrow$

$i \downarrow$

		1	2	-	d	-
i	1	0	0	-		-
	2	0	0	-		-
$\downarrow$	-	-	-	-		-
	d				0	
	-	-	-	-		-

$(d, d) = 0 \Rightarrow w_d \in L_d$

TS M_d accepts L_d

$(d, d) = 1$

Uncomputability of a Diagonal Language

- **TM M_d that accepts the diagonal language $L_d = L(M_d)$**

$j \rightarrow$

$i \downarrow$	1	2	-	d	-
1	0	0	-		-
2	0	0	-		-
-	-	-	-		-
d				1	
-	-	-	-		-

$(d, d) = 1 \Rightarrow w_d \notin L_d$

TM M_d accepts L_d

$(d, d) = 0$

Lecture 14

4.2 UNRESTRICTED GRAMMAR

- 4.2.1 Construction of a Turing Machine from an Unrestricted Grammar
- 4.2.2 Construction of a Grammar from a Turing Machine

4.3 PROPERTIES OF RECURSIVE AND RECURSIVELY ENUMERABLE (RE) LANGUAGES

- 4.3.1 Closure properties
- 4.3.2 Computability
- 4.3.3 Decidability

Decidability

- **A problem is decidable**
  - if there exists a TM that always halts and decides about the input aceptance
- **Recursive languages**
  - DECIDABLE
  - COMPUTABLE
- **Recursively enumerable languages**
  - Not DECIDABLE
  - COMPUTABLE

Universal Turing Machine

Code < TS *M*, *w* >	111 $code_1$ 11 $code_2$ 11 - - - 11 $code_r$ 111 *w*

Tape of TM *M*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*

State of TM *M*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*

Universal Turing Machine

Code $< \text{TS } M, w >$	$111\ q_{l1}X_{l1}q_{d1}X_{d1}D_{d1}\ 11\ q_{l2}X_{l2}q_{d2}X_{d2}D_{d2}\ 111\ w$														
Tape of* TM *M		*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*
State of* TM *M	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*

Universal Turing Machine

Code < TS *M*, *w* >

111 $q_{l1}X_{l1}q_{d1}X_{d1}D_{d1}$ 11 $q_{l2}X_{l2}q_{d2}X_{d2}D_{d2}$ 111 w

Tape of* TM *M

w	B	B	B	B	B	B	B	B	B	B

State of* TM *M

q_1	B	B	B	B	B	B	B	B	B	B	B	B

Universal Turing Machine

$q_{l2}\, X_{l2}$

Code < TS *M*, *w* > | 111 $q_{l2} X_{l2} q_{d1} X_{d1} D_{d1}$ 11 $q_{l2} X_{l2} q_{d2} X_{d2} D_{d2}$ 111 *w*

Tape of TM *M*	β X_{l2} α	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*		
State of TM *M*	q_{l2}	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*

Universal Turing Machine

$q_{l2}\ X_{l2}$

Code < TS M, w >	111 $q_{l1} X_{l1} q_{d1} X_{d1} D_{d1}$ 11 $q_{l2} X_{l2} q_{d2} X_{d2} D_{d2}$ 111 w

Tape of TM M	$\beta \quad X_{l2} \quad \alpha$	B	B	B	B	B	B	B	B	B	B

State of TM M	q_{l2}	B	B	B	B	B	B	B	B	B	B	B	B

Universal Turing Machine

Code < TS *M*, *w* > $111\ q_{l1}X_{l1}q_{d1}X_{d1}D_{d1}\ 11\ q_{l2}X_{l2}q_{d2}X_{d2}D_{d2}\ 111\ w$

Tape of* TM *M	$\beta\ X_{d2}\ \alpha$	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*

State of* TM *M	q_{d2}	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*

Universal Turing Machine

Code < TS *M*, *w* > 111 $q_{l1}X_{l1}q_{d1}X_{d1}D_{d1}$ 11 $q_{l2}X_{l2}q_{d2}X_{d2}D_{d2}$ 111 w

Tape of* TM *M

β	X_{d2}	α	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*

State of* TM *M

q_{d2}	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*

Universal TM M_u *acccepts the universal language* L_u

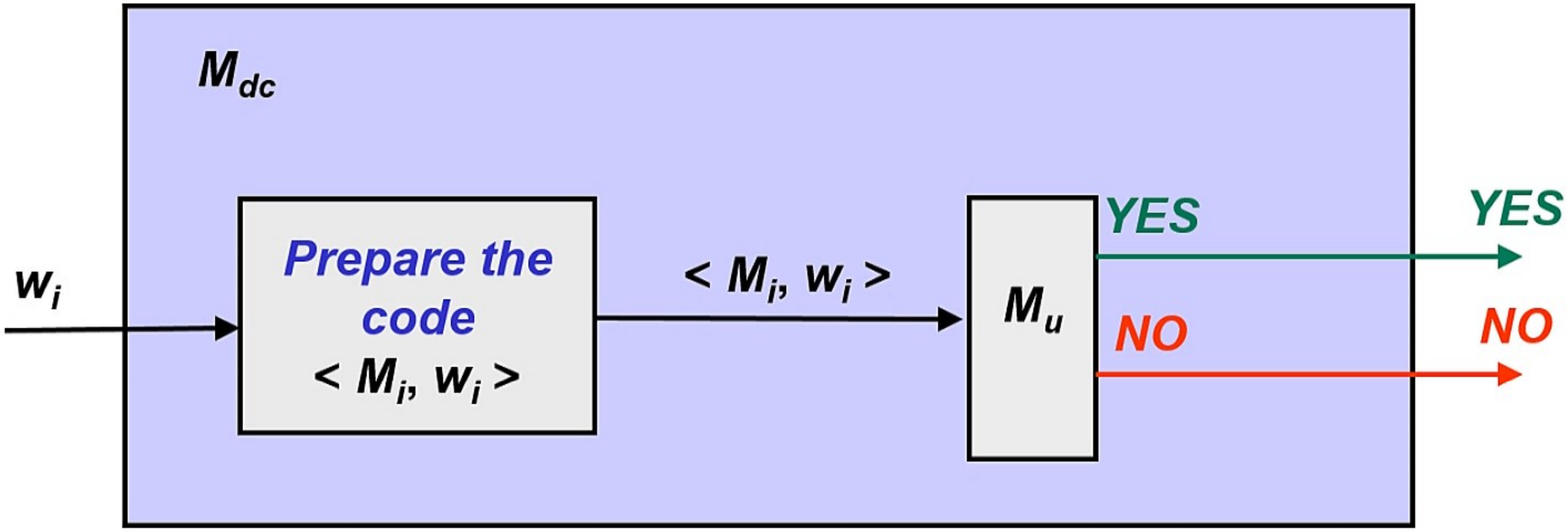
L_u = {< *M*, *w* > | *TM M accepts the string w* }

Universal Language

- **Universal language *L_u***
  - recursively enumerable – computable
  - accepted by a universal Turing machine TM M_u

  - not recursive – not decidable
  - ASSUMPTION – Universal language L_u is decidable

Universal Language



- M_u **always halts** $\Rightarrow$ M_{dc} **always halts** $\Rightarrow$ L_d^c **is recursive** $\Rightarrow$ $(L_d^c)^c = L_d$ **is recursive**
- **Before we proved that**
  - **diagonal language is not recursively enumerable**
    - — **TM that accepts the diagonal language DOES NOT EXIST**
  - **diagonal language is not recursive**
- **Initial assuption that universal language is decidable IS WRONG**
  - **leads to a wrong conclusion that diagonal language is recursive**