Order of invention:

1. Regular languages and FAs were developed by Kleene, Mealy, Moore, Rabin, and Scott in the 1950s.
2. CFGs and PDAs were developed later, by Chomsky, Gettinger, Schtitzenberger, and Evey, mostly in the 1960s.
3. Turing machines and their theory were developed by Alan Mathison Turing and Emil Post in the 1930s and 1940s.

Language Defined by	Corresponding Acceptor	Nondeterminism = Determinism?	Language Closed Under	What Can Be Decided	Example of Application
Regular expression	Finite automaton, transition graph	Yes	Union, product, Kleene star, intersection, complement	Equivalence, emptiness, finiteness, membership	Text editors, sequential circuits
Context-free grammar	Pushdown automaton	No	Union, product, Kleene star	Emptiness finiteness membership	Programming language statements, compilers
Type 0 grammar	Turing machine, Post machine, 2PDA, *n*PDA	Yes	Union, product, intersection, Kleene star	Not much	**Computers**

Turing Machines:

A Turing machine (TM) consists of the following

1. An alphabet Σ of input letters.
2. An input TAPE partitioned into cells, having infinite many locations in one direction. The input string is placed on the TAPE starting its first letter on the cell i, the rest of the TAPE is initially filled with blanks (Δ's).

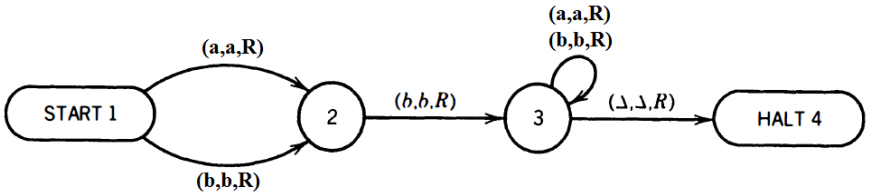
a	b	a	a	a	Δ	Δ	Δ
i	ii	iii	iv	v	vi	vii	

3. A tape Head can read the contents of cell on the TAPE in one step. It can replace the character at any cell and can reposition itself to the next cell to the right or to the left of that it has just read. Initially the TAPE Head is at the cell i. The TAPE Head can't move to the left of cell i. The location of the TAPE Head is denoted by ⌂.
4. An alphabet Γ of characters that can be printed on the TAPE by the TAPE Head. Γ may include the letters of Σ. Even the TAPE Head can print blank Δ, which means to erase some character from the TAPE.
5. Finite set of states containing exactly one START state and some (may be none) HALT states that cause execution to terminate when the HALT states are entered.
6. A program which is the set of rules, which shows that which state is to be entered when a letter is read from the TAPE and what character is to be printed. This program is shown by the states connected by directed edges labeled by triplet (letter, letter, direction). It may be noted that the first letter is the character the TAPE Head reads from the cell to which it is pointing. The second

letter is what the TAPE Head prints the cell before it leaves. The direction tells the TAPE Head whether to move one cell to the right, R (Move Forward), or one cell to the left, L (Move Backward).

Note:

- There may not be any outgoing edge at a certain state for a certain letter to be read from the TAPE, which creates non-determinism in Turing machines.
- It may also be noted that at a certain state, there can't be more than one out going edges for a certain letter to be read from the TAPE.
- The machine crashes if there is no path for a letter to be read from the TAPE and the corresponding string is supposed to be rejected.
- To terminate execution of a certain input string successfully, a HALT state must be entered and the corresponding string is supposed to be accepted by the TM.
- The machine also crashes when the TAPE Head is instructed to move one cell to the left of cell i.



RE: (a+b)b(a+b)* String: aba

State #1: START
READ – a
WRITE – a
Head: Move Forward
Goto State #2

a	b	a	Δ	Δ	Δ	Δ	Δ
i	ii	iii	iv	v	vi	vii	

a	b	a	Δ	Δ	Δ	Δ	Δ
i	ii	iii	iv	v	vi	vii	

State #2
READ – b
WRITE – b
Head: Move Forward
Goto State #3

a	b	a	Δ	Δ	Δ	Δ	Δ
i	ii	iii	iv	v	vi	vii	

a	b	a	Δ	Δ	Δ	Δ	Δ
i	ii	iii	iv	v	vi	vii	

State #3
READ – a
WRITE – a
Head: Move Forward
Goto State #3

a	b	a	Δ	Δ	Δ	Δ	Δ
i	ii	iii	iv	v	vi	vii	

a	b	a	Δ	Δ	Δ	Δ	Δ
i	ii	iii	iv	v	vi	vii	

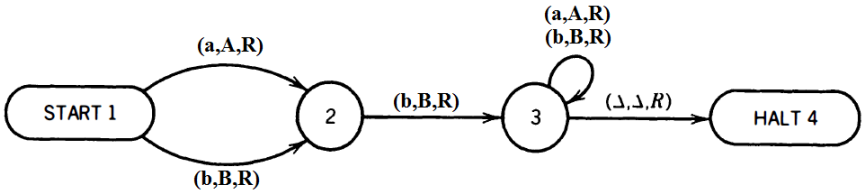
State #3
READ – Δ
WRITE – Δ
Head: Move Forward
Goto State #4: HALT

a	b	a	Δ	Δ	Δ	Δ	Δ
i	ii	iii	iv	v	vi	vii	

a	b	a	Δ	Δ	Δ	Δ	Δ
i	ii	iii	iv	v	vi	vii	

So, String is ACCEPTED
Tape:

a	b	a	Δ	Δ	Δ	Δ	Δ
i	ii	iii	iv	v	vi	vii	



String: aba

State #1: START
READ – a
WRITE – A
Head: Move Forward
Goto State #2

a	b	a	Δ	Δ	Δ	Δ	Δ
i	ii	iii	iv	v	vi	vii	

A	b	a	Δ	Δ	Δ	Δ	Δ
i	ii	iii	iv	v	vi	vii	

State #2
READ – b
WRITE – B
Head: Move Forward
Goto State #3

A	b	a	Δ	Δ	Δ	Δ	Δ
i	ii	iii	iv	v	vi	vii	

A	B	a	Δ	Δ	Δ	Δ	Δ
i	ii	iii	iv	v	vi	vii	

State #3
READ – a
WRITE – A
Head: Move Forward
Goto State #3

A	B	a	Δ	Δ	Δ	Δ	Δ
i	ii	iii	iv	v	vi	vii	

A	B	A	Δ	Δ	Δ	Δ	Δ
i	ii	iii	iv	v	v	vii	

State #3
READ – Δ
WRITE – Δ
Head: Move Forward
Goto State #4: HALT

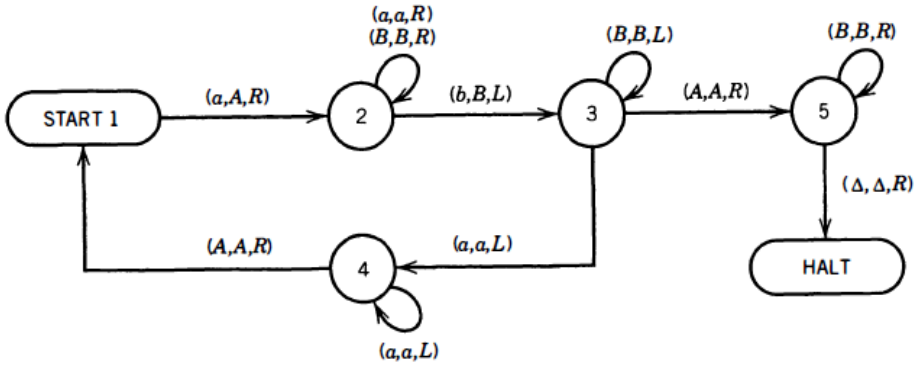
A	B	A	Δ	Δ	Δ	Δ	Δ
i	ii	iii	iv	v	vi	vii	

A	B	A	Δ	Δ	Δ	Δ	Δ
i	ii	iii	iv	v	vi	vii	

So, String is ACCEPTED
Tape:

A	B	A	Δ	Δ	Δ	Δ	Δ
i	ii	iii	iv	v	vi	vii	

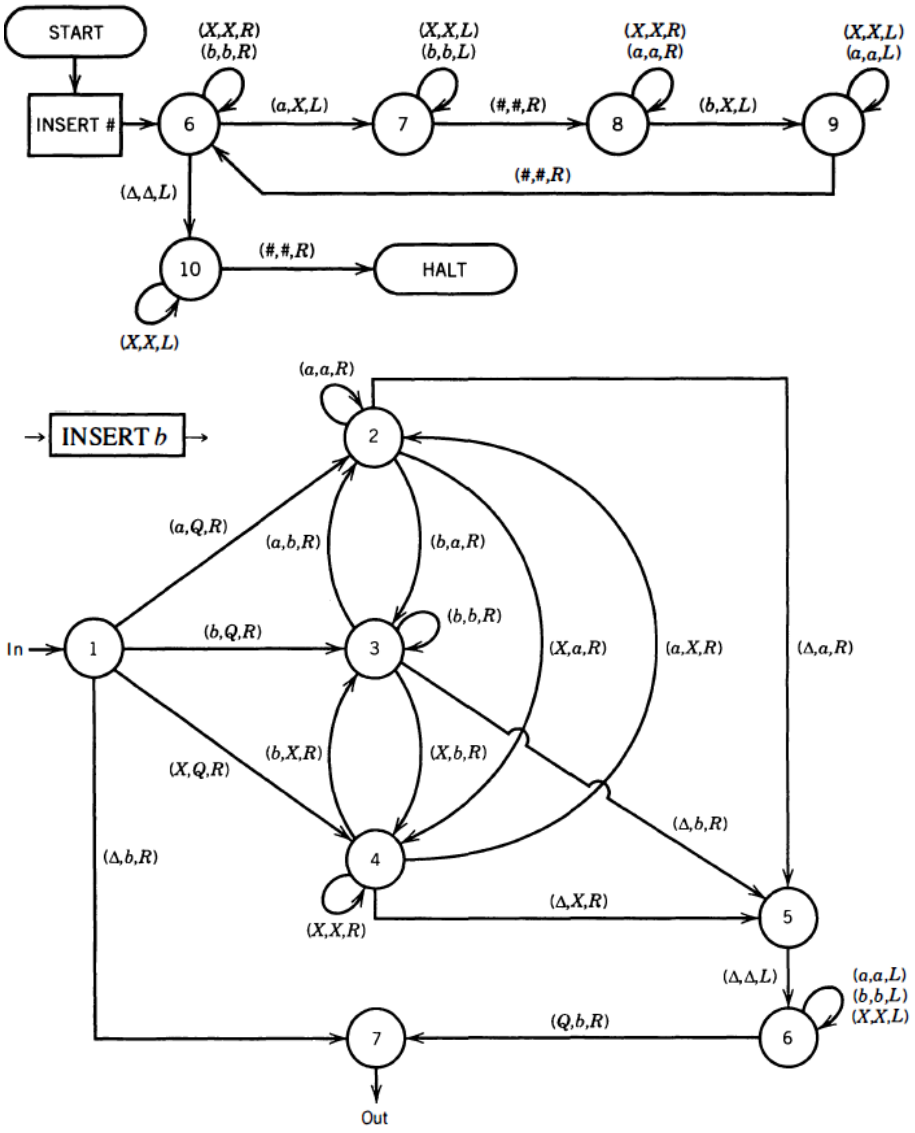
EXAMPLE:

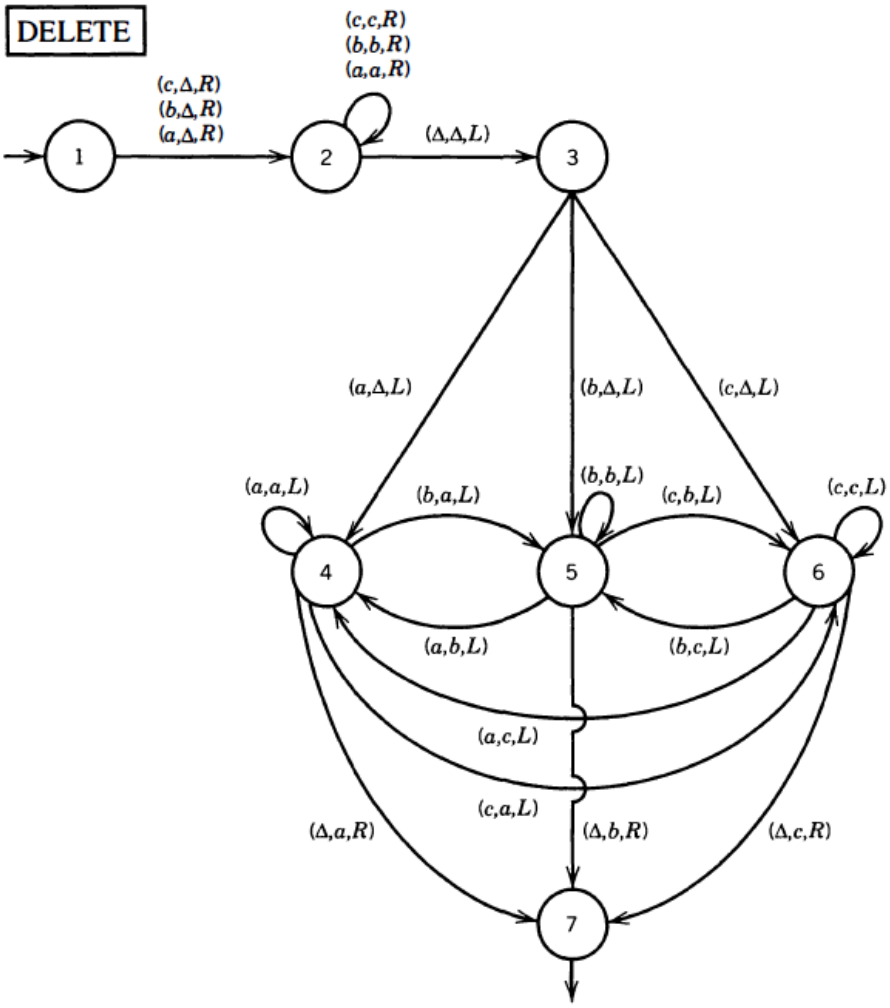


State	Tape	Read & Write	Direction	State	Tape
START 1	aabbΔ	a, A	R	2	Aabbb Δ
2	Aabbb Δ	a,a	R	2	Aabbb Δ
2	Aabbb Δ	b, B	L	3	AaBbΔ
3	AaBbΔ	a, a	L	4	AaBbΔ
4	AaBbΔ	A, A	R	1	AaBbΔ
1	AaBbΔ	a, A	R	2	AABbΔ
2	AABbΔ	B, B	R	2	AABbΔ
2	AABbΔ	b, B	L	3	AABBΔ
3	AABBΔ	B, B	L	3	AABBΔ
3	AABBΔ	A, A	R	5	AABBΔ
5	AABBΔ	B, B	R	5	AABBΔ
5	AABBΔ	B, B	R	5	AABBΔ
5	AABBΔ	Δ, Δ	R	HALT	AABBΔΔ

$$\underset{1}{\underline{a}abb} \rightarrow \underset{2}{A\underline{a}bb} \rightarrow \underset{2}{Aa\underline{b}b} \rightarrow \underset{3}{A\underline{a}Bb} \rightarrow \underset{4}{\underline{A}aBb} \rightarrow \underset{1}{A\underline{a}Bb}$$
$$\rightarrow \underset{2}{AA\underline{B}b} \rightarrow \underset{2}{AAB\underline{b}} \rightarrow \underset{3}{AA\underline{B}B} \rightarrow \underset{3}{A\underline{A}BB} \rightarrow \underset{5}{AA\underline{B}B} \rightarrow \underset{5}{AAB\underline{B}}$$
$$\rightarrow \underset{5}{AABB\underline{\Delta}} \rightarrow \text{HALT}$$

EXAMPLE:





Task: Test string abbc on the DELETE sub-TM.