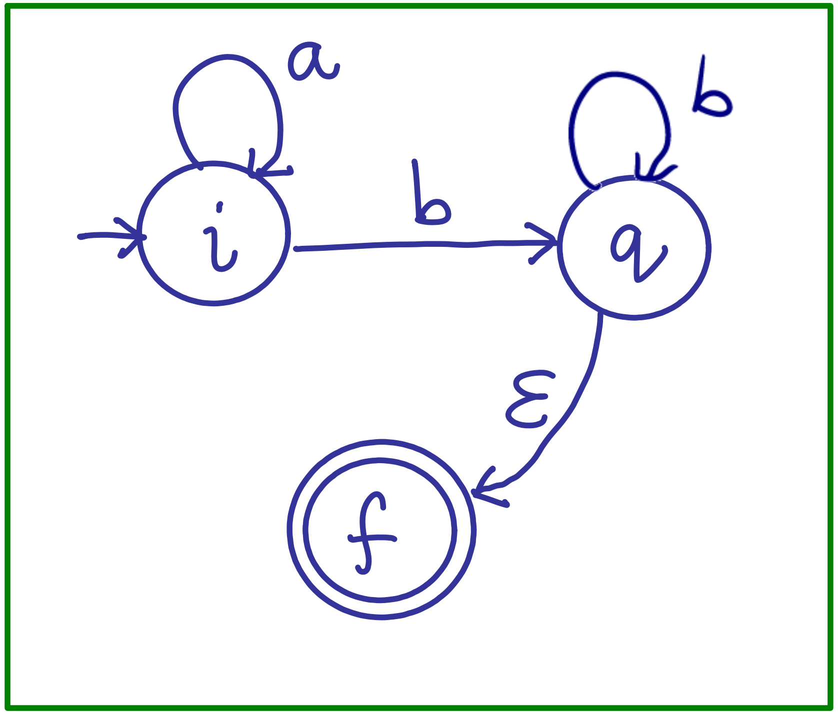
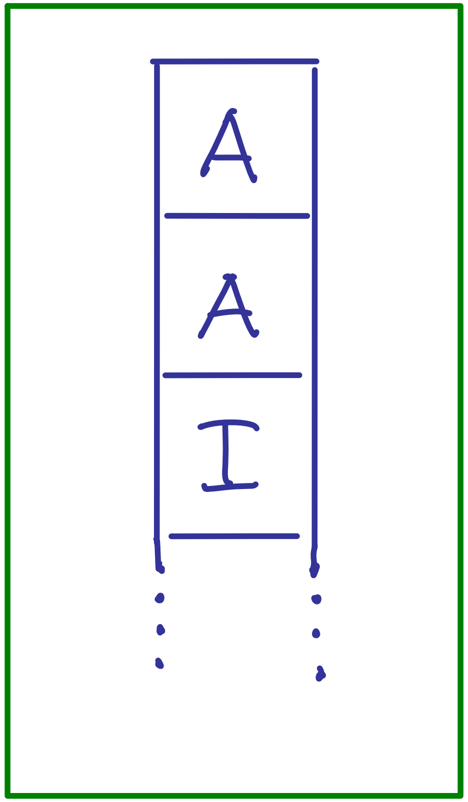
Pushdown automaton - informally



+



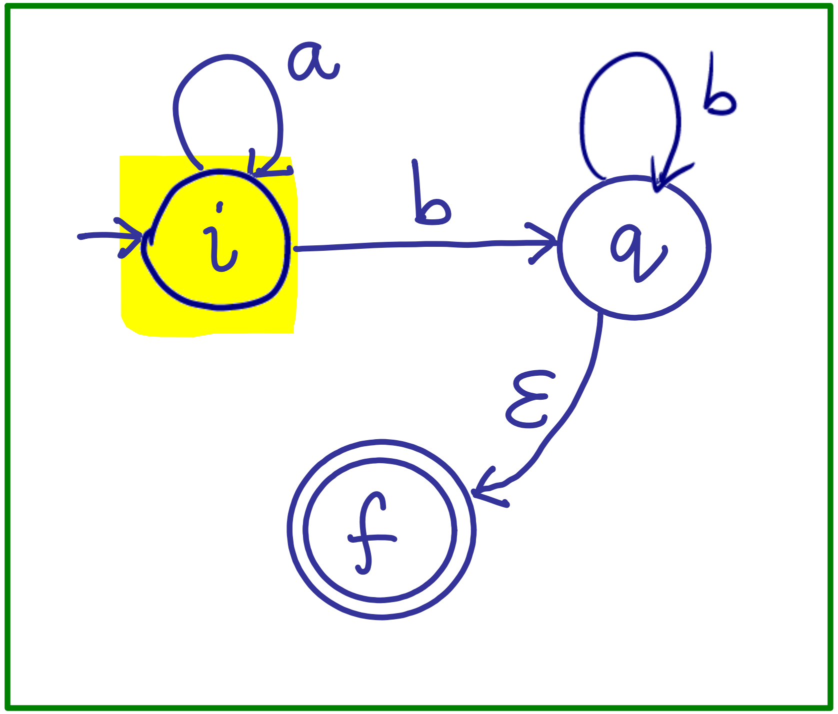
CONTROL (finite state machine)

MEMORY (stack)

aaabbb

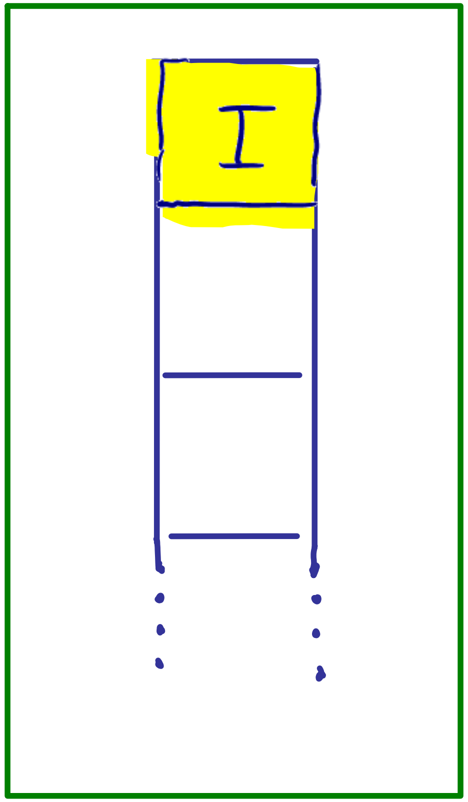
INPUT

Pushdown automaton - informally



CONTROL (finite state machine)

+

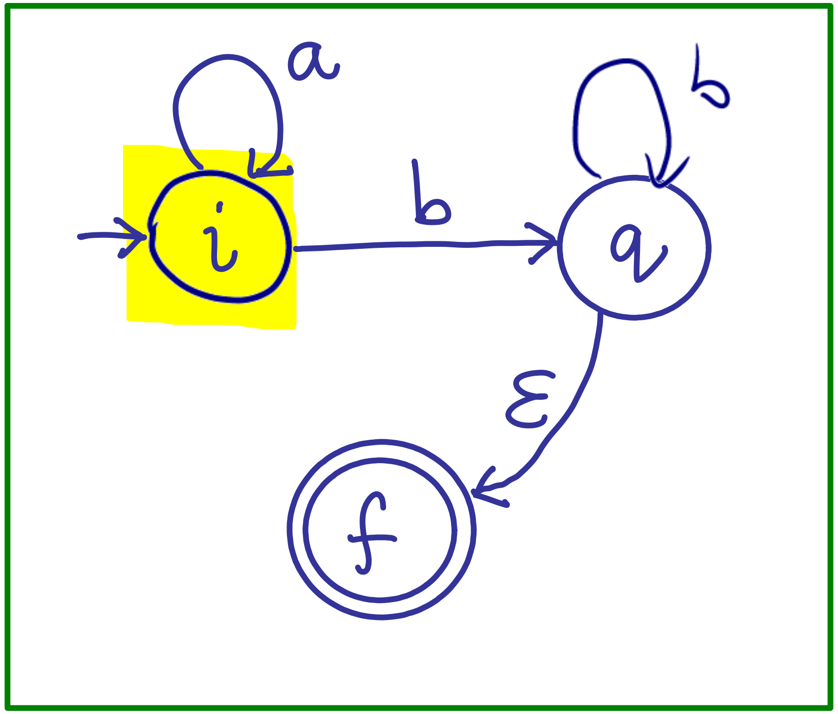


MEMORY (stack)

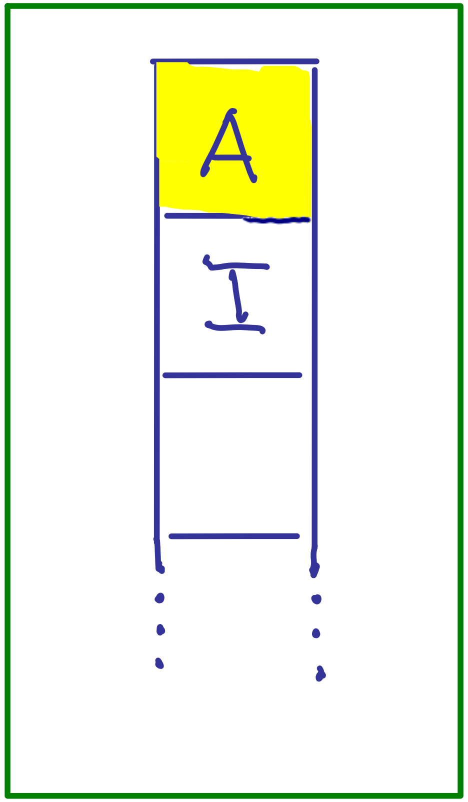
aaaabbb

INPUT

Pushdown automaton – informally



+



push A

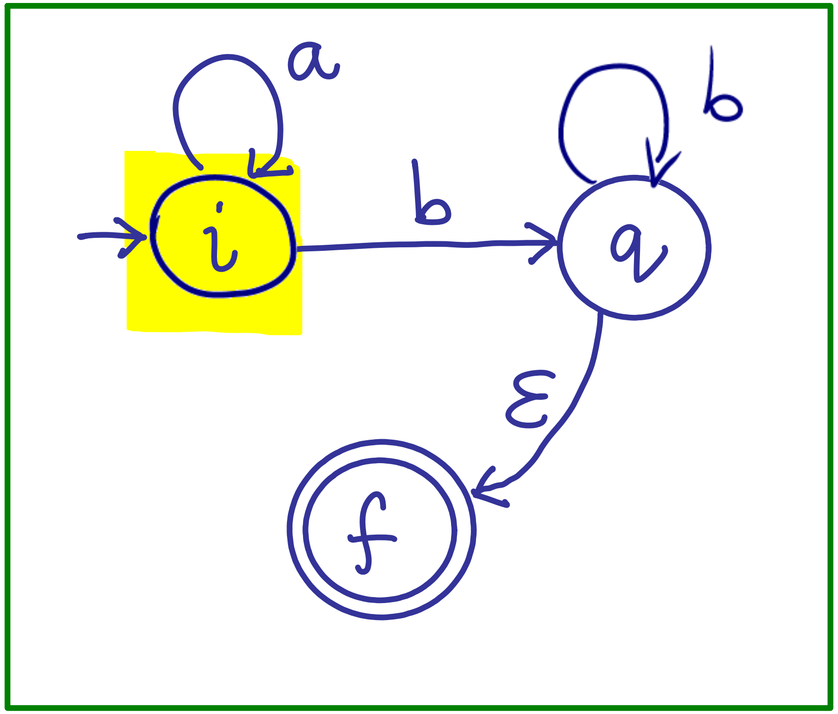
CONTROL (finite state machine)

MEMORY (stack)

~~a~~aabbb

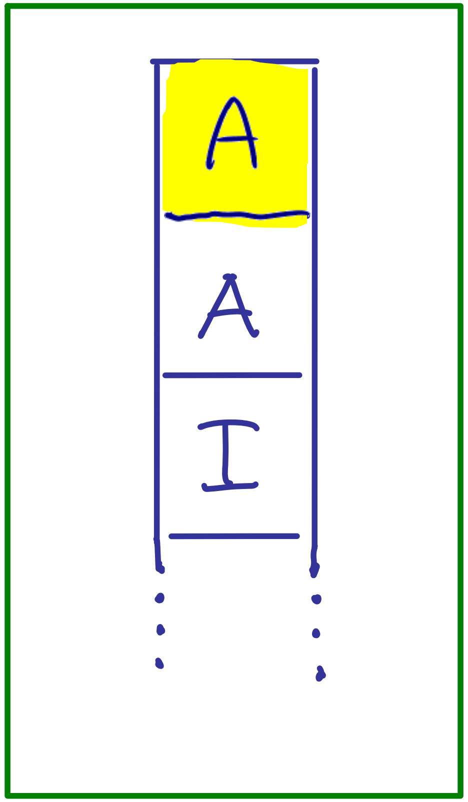
INPUT

Pushdown automaton – informally



CONTROL (finite state machine)

+



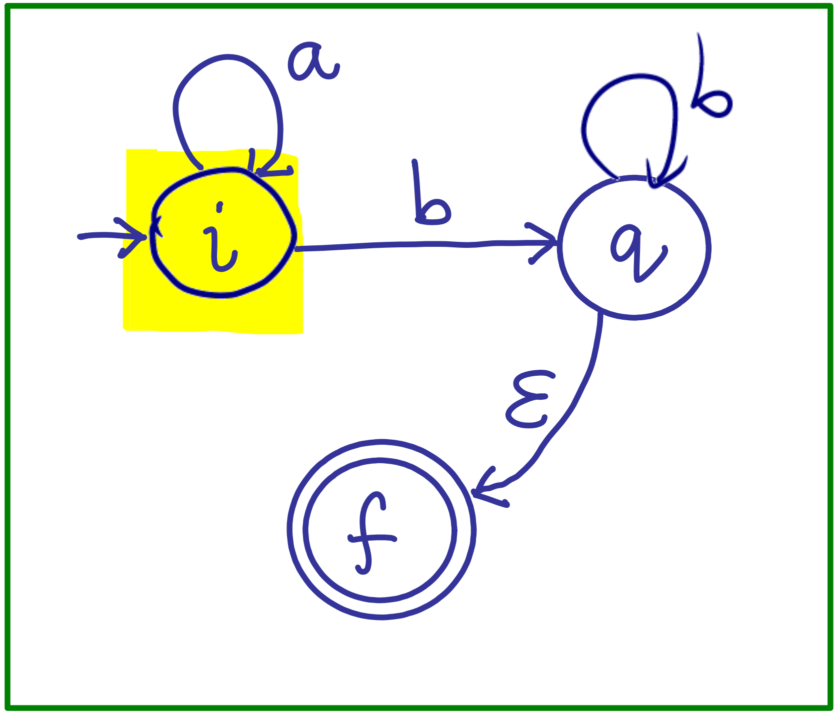
MEMORY (stack)

push A

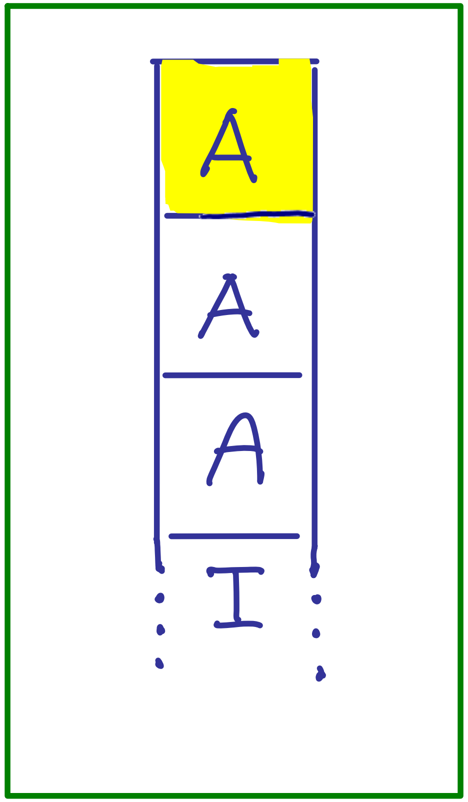
~~a~~ ~~a~~ a b b b

INPUT

Pushdown automaton – informally



+



push A

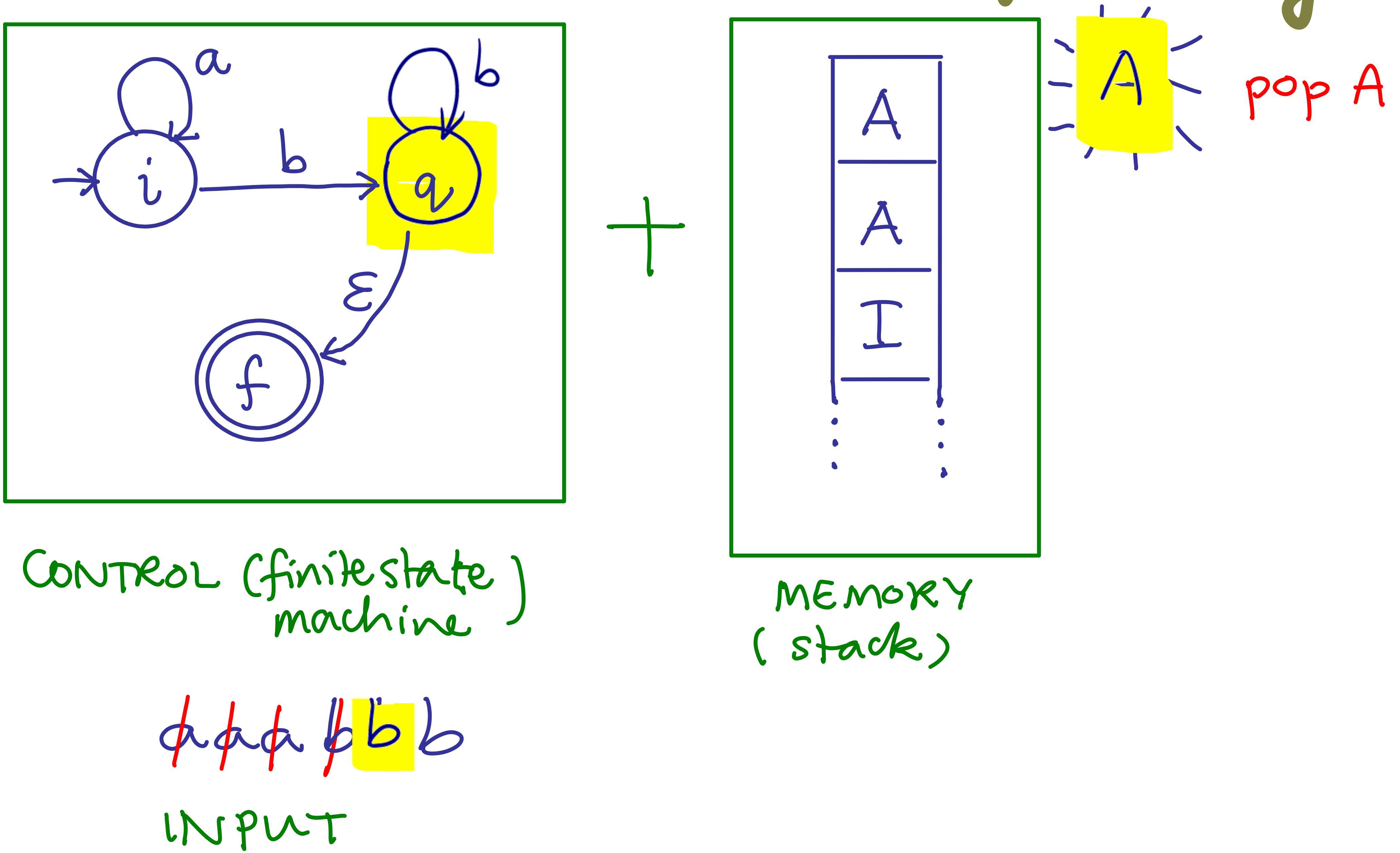
CONTROL (finite state machine)

MEMORY (stack)

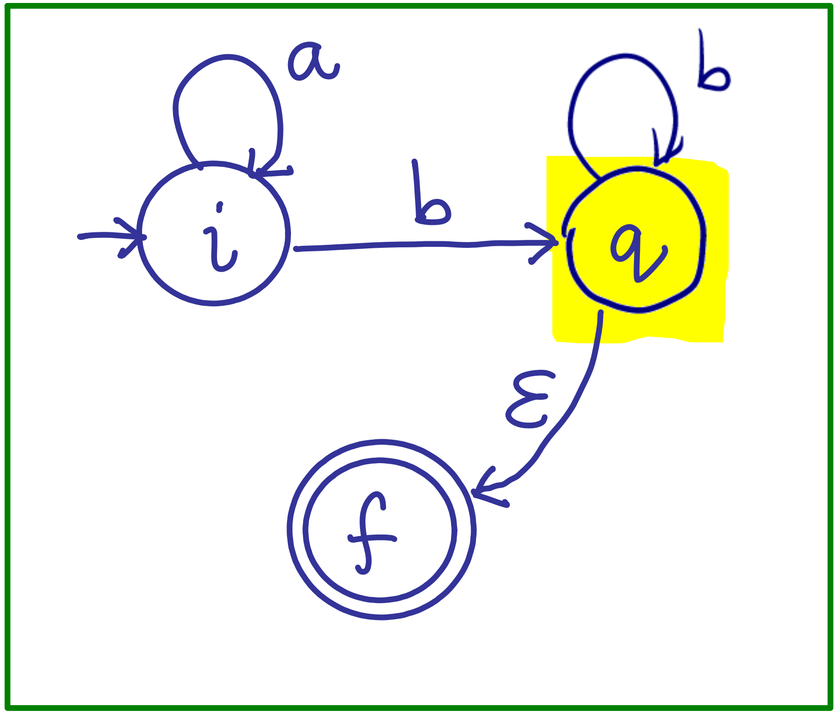
~~a~~~~a~~~~a~~bbb

INPUT

Pushdown automaton - informally

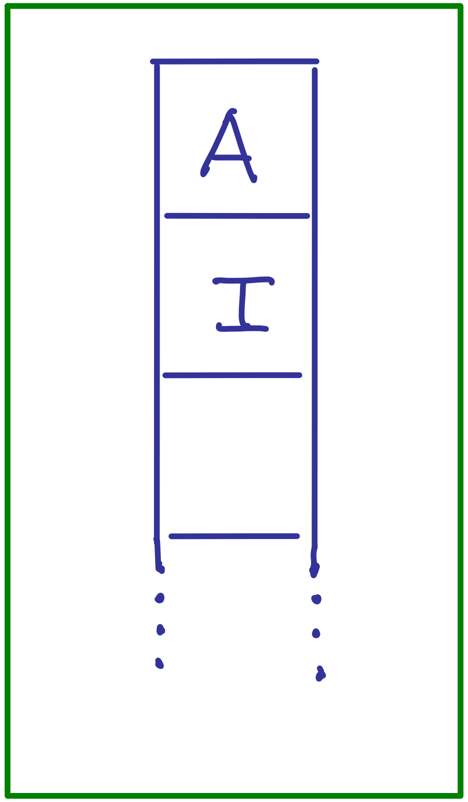


Pushdown automaton – informally

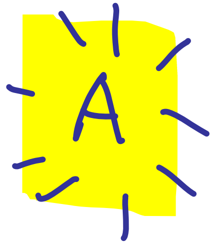


CONTROL (finite state machine)

+



MEMORY (stack)

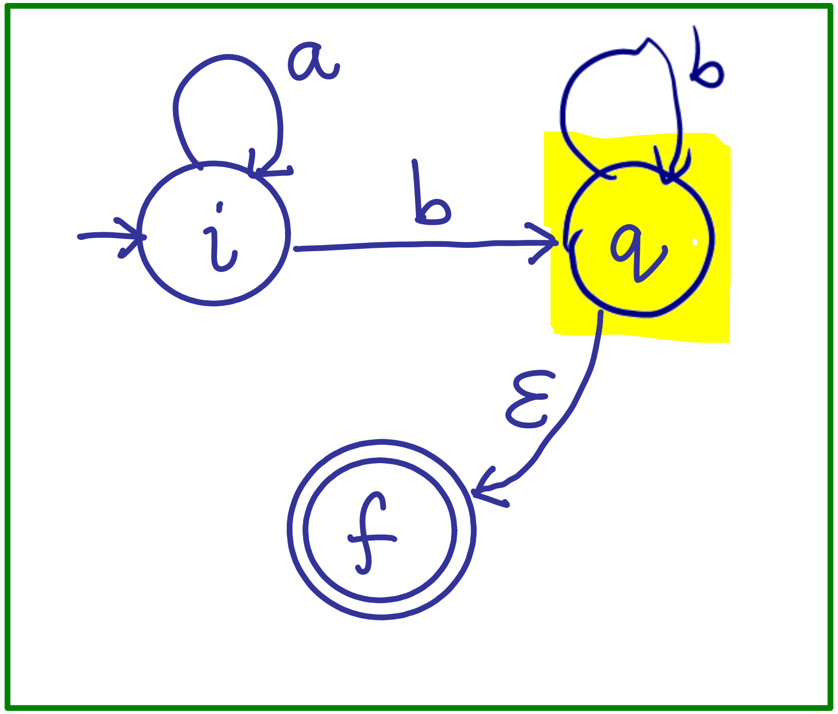


pop A

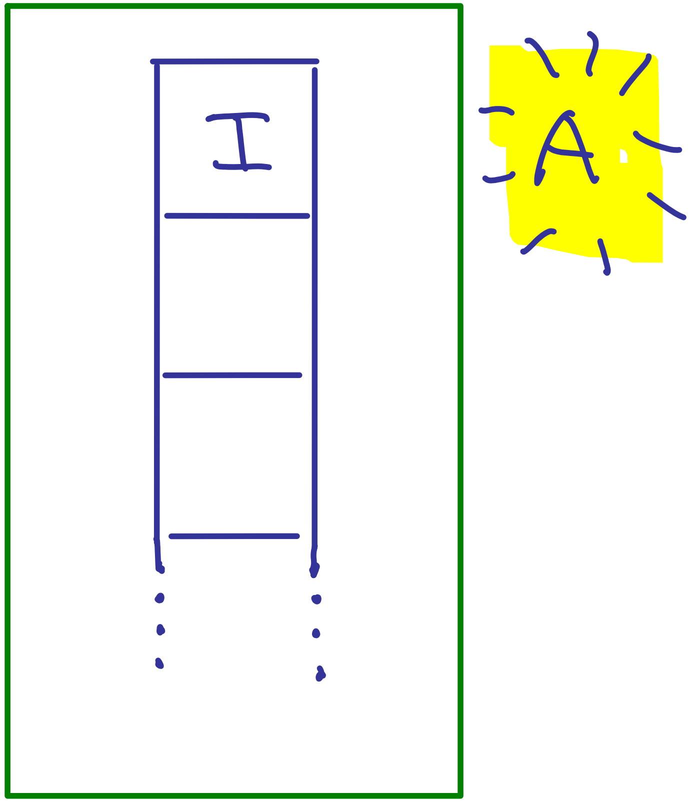
$\cancel{a}\cancel{a}\cancel{a}\cancel{b}\cancel{b}b$

INPUT

Pushdown automaton – informally



+



pop A

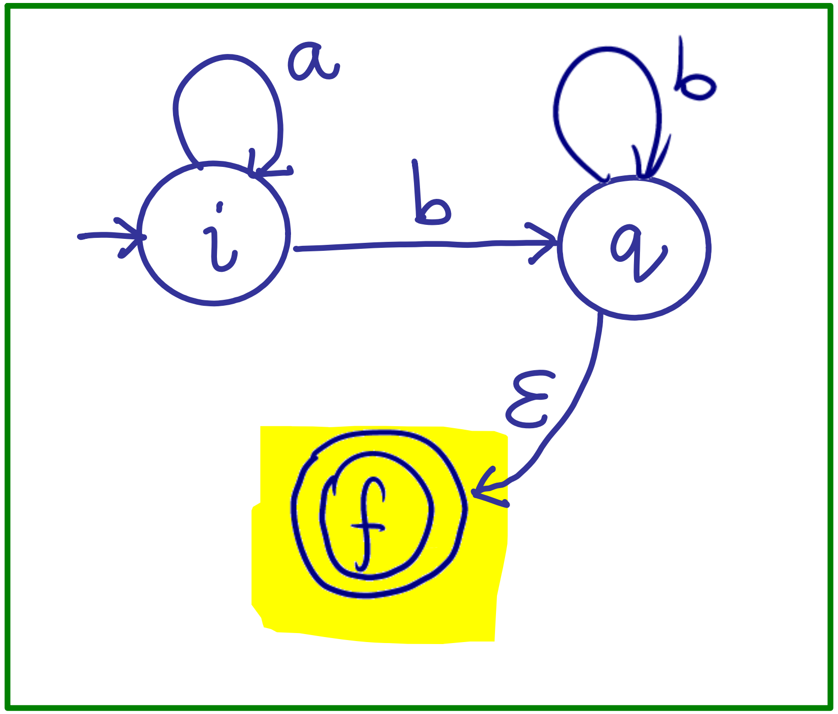
CONTROL (finite state machine)

MEMORY (stack)

~~a~~ ~~a~~ ~~a~~ ~~b~~ ~~b~~ ~~b~~

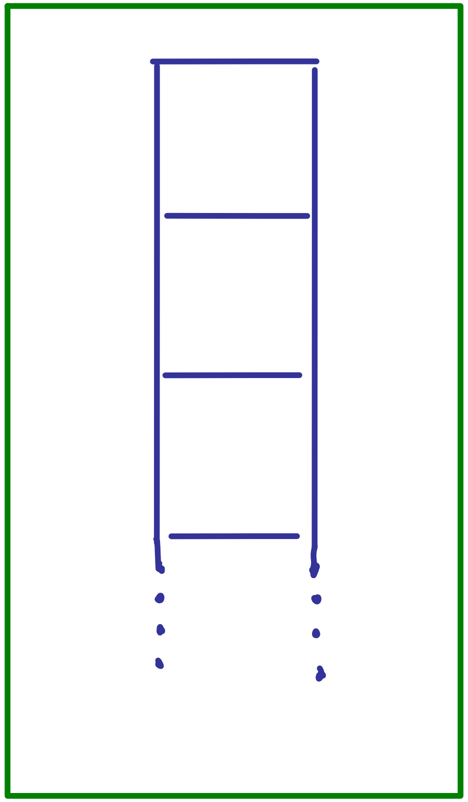
INPUT

Pushdown automaton - informally

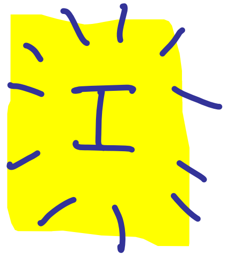


CONTROL (finite state machine)

+



MEMORY (stack)



pop I

~~aaabbb~~

INPUT

Non-deterministic Pushdown Automaton (NPDA)

is specified by:

- Q, finite set of machine ***states***
- Σ, alphabet of ***input symbols***
- $s \in Q$, the ***start state***
- $F \subseteq Q$, subset of ***accepting states***
- Γ, alphabet of ***stack symbols***
- $I \in \Gamma$, the ***initial stack symbol***
- Δ, finite set of ***transitions***, which are either ***input-transitions*** $\boxed{A, q \xrightarrow{a} S, q'}$, or ε***-transitions*** $\boxed{A, q \xrightarrow{\varepsilon} S, q'}$ (where $A \in \Gamma$, $q \in Q$, $a \in \Sigma$, $S \in \Gamma^*$ and $q' \in Q$).

Allowed operations on stacks $S \in \Gamma^*$

pop the top element A off a non-empty stack AS, producing a new stack S and returning the element A

push a finite string S' of elements on to the top of a stack S, producing a new stack $S'S$

Note:

- pop is not defined on the empty stack;
- we may push an empty string onto a stack (in which case it is unchanged).

Example NPDA

States: $\boldsymbol{i} \quad \boldsymbol{q} \quad \boldsymbol{f}$

Input symbols: $\boldsymbol{a} \quad \boldsymbol{b}$

Start state: $\boldsymbol{i}$

Accepting state: $\boldsymbol{f}$

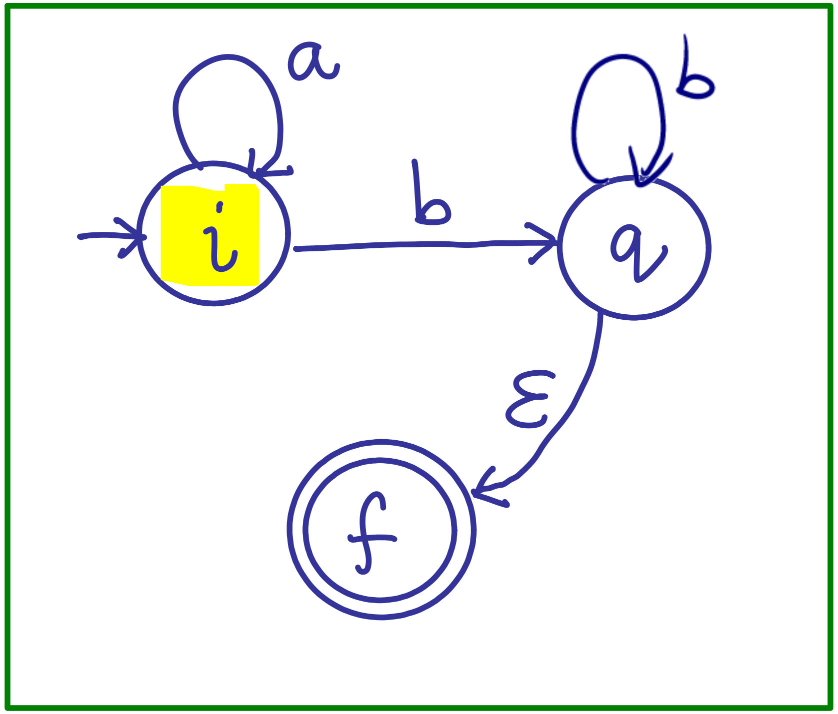
Stack symbols: $\boldsymbol{I} \quad \boldsymbol{A}$

Initial stack symbol: $\boldsymbol{I}$

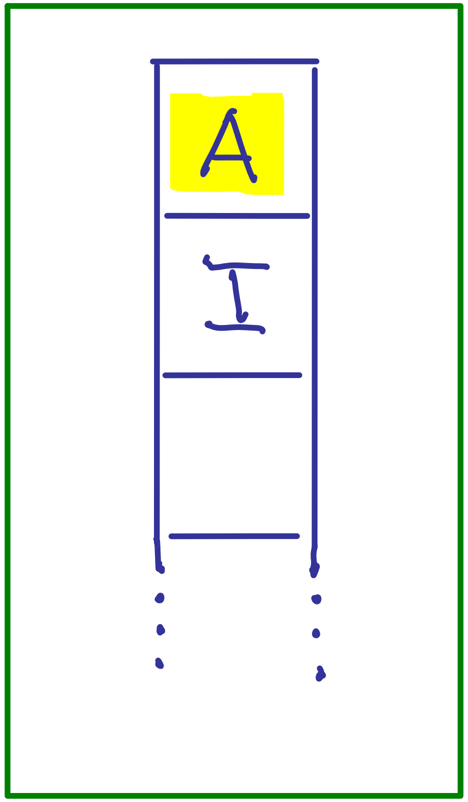
Transitions:

ε-transition	a-transitions	b-transitions
$I, q \xrightarrow{\varepsilon} \varepsilon, f$	$I, i \xrightarrow{a} AI, i$	$A, i \xrightarrow{b} \varepsilon, q$
	$A, i \xrightarrow{a} AA, i$	$A, q \xrightarrow{b} \varepsilon, q$

Pushdown automaton – informally



+



push A

CONTROL (finite state machine)

MEMORY (stack)

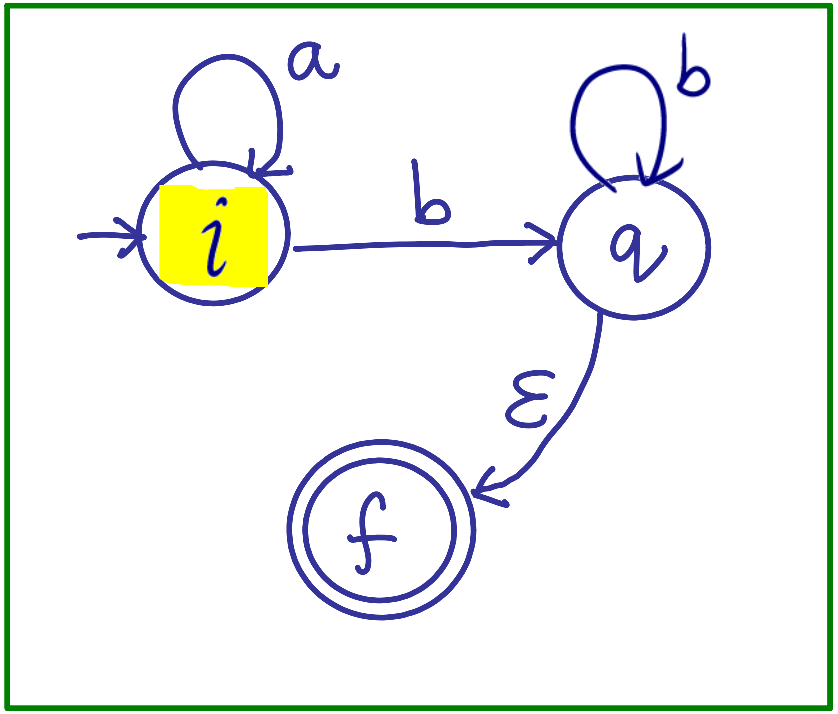
~~a~~aabbb

INPUT

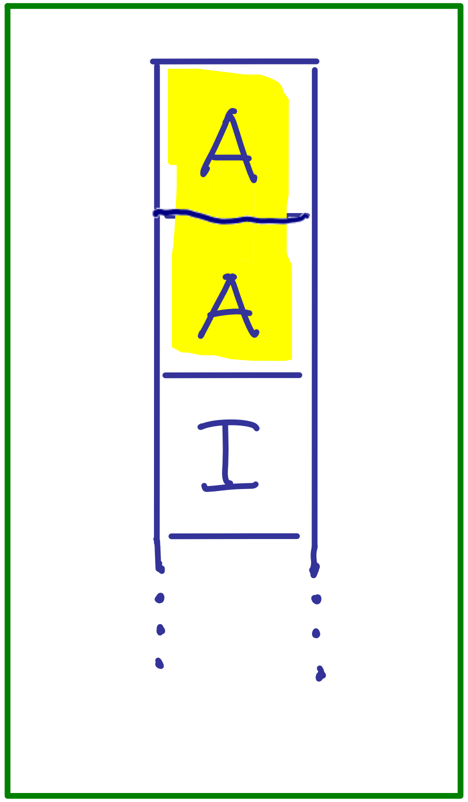
input transition

$A, i \xrightarrow{a} AA, i$

Pushdown automaton – informally



+



push A

CONTROL (finite state machine)

MEMORY (stack)

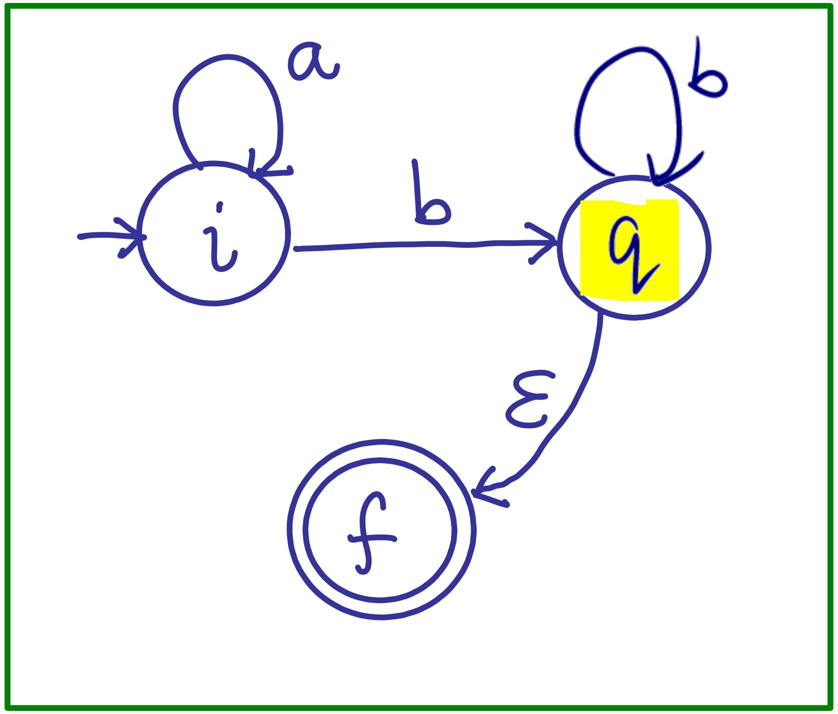
~~a~~ ~~a~~ a a b b b

INPUT

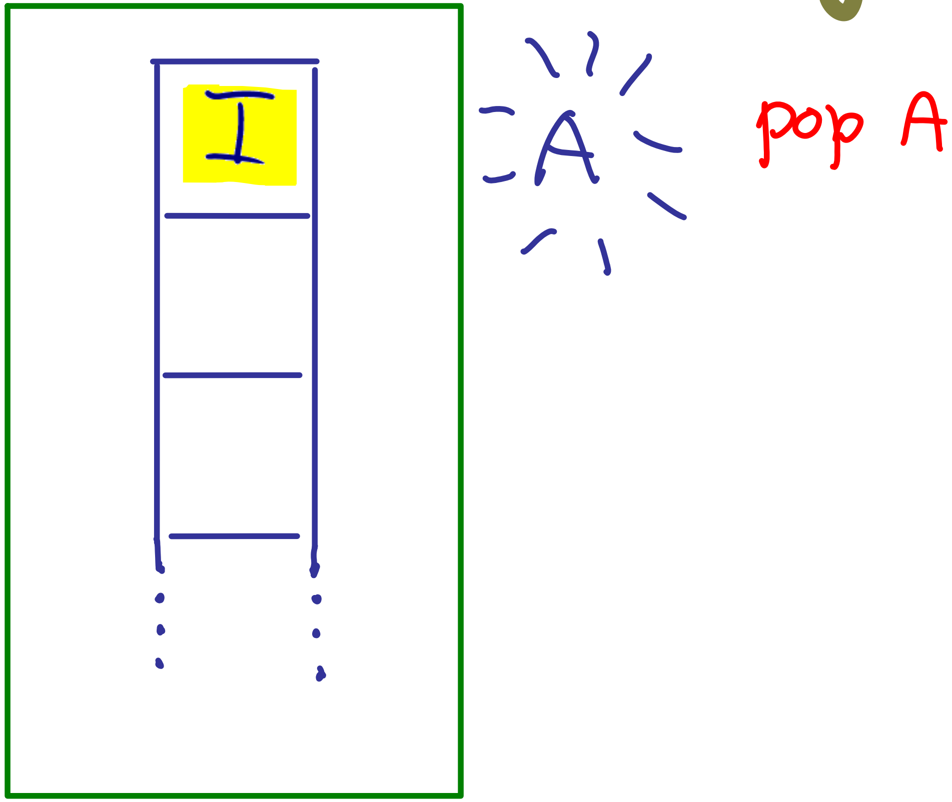
input transition

$A, i \xrightarrow{a} \boxed{AA}, \boxed{i}$

Pushdown automaton - informally



+



CONTROL (finite state machine)

MEMORY (stack)

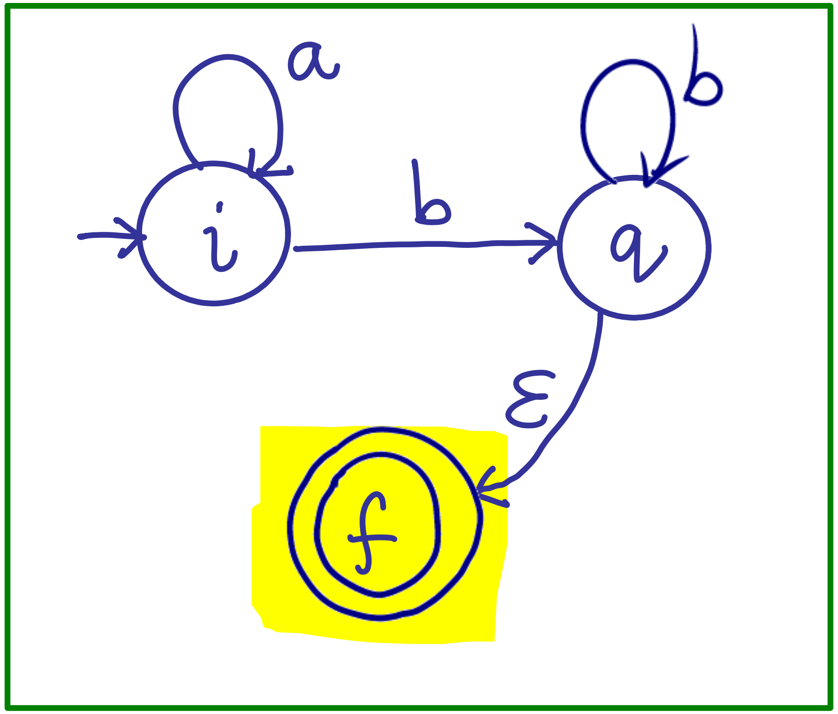
~~aaabbb~~

INPUT

ε-transition

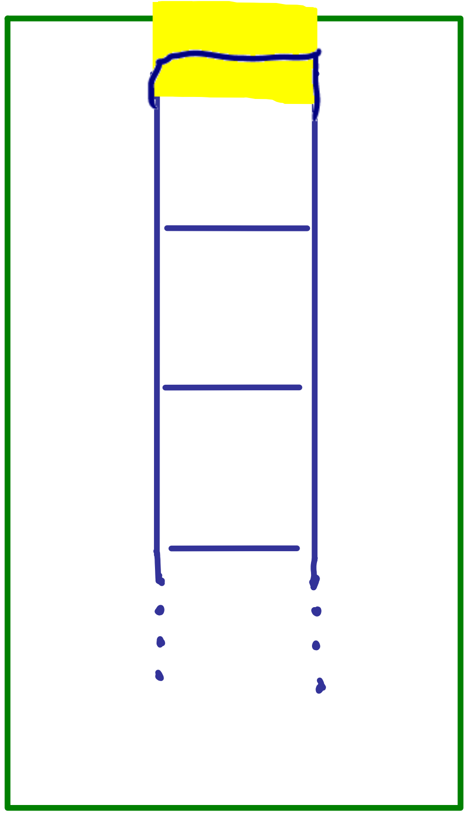
$I, q \xrightarrow{\varepsilon} \varepsilon, f$

Pushdown automaton – informally



CONTROL (finite state machine)

+



MEMORY
(stack)

I

pop I

$\cancel{a}\cancel{a}\cancel{a}\cancel{b}\cancel{b}\cancel{b}$

INPUT

ε-transition

$I, q \xrightarrow{\varepsilon} \varepsilon, f$

Next-configuration relation $(S, q, w) \Rightarrow^1 (S', q', w')$

describes how a NDPA $M = (Q, \Sigma, s, F, \Gamma, I, \Delta)$ can move from one configuration (S, q, w) to another (S', q, w') in one step.

It is defined to hold if it matches either of the following two cases:

- $(AS', q, aw) \Rightarrow^1 (SS', q', w)$
  where $A, q \xrightarrow{a} S, q'$ is an input-transition of M;
- $(AS', q, w) \Rightarrow^1 (SS', q', w)$
  where $A, q \xrightarrow{\varepsilon} S, q'$ is an ε-transition of M.

We write $\boxed{(S, q, w) \Rightarrow^* (S', q', w')}$ to mean

$$(S, q, w) = (S_1, q_1, w_1) \Rightarrow^1 \cdots \Rightarrow^1 (S_n, q_n, w_n) = (S', q', w')$$

holds for some $n \geq 1$ and configurations (S_i, q_i, w_i).

$(I, i, a^3b^3) \Rightarrow^1 (AI, i, a^2b^3)$ by $I, i \xrightarrow{a} AI, i$

$$(I, i, a^3b^3) \Rightarrow^1 (AI, i, a^2b^3) \qquad \text{by } I, i \xrightarrow{a} AI, i$$

$$\Rightarrow^1 (AAI, i, ab^3) \qquad \text{by } A, i \xrightarrow{a} AA, i$$

$$\Rightarrow^1 (AAAI, i, b^3) \qquad \text{by } A, i \xrightarrow{a} AA, i$$

$$(I, i, a^3b^3) \Rightarrow^1 (AI, i, a^2b^3) \quad \text{by } I, i \xrightarrow{a} AI, i$$

$$\Rightarrow^1 (AAI, i, ab^3) \quad \text{by } A, i \xrightarrow{a} AA, i$$

$$\Rightarrow^1 (AAAI, i, b^3) \quad \text{by } A, i \xrightarrow{a} AA, i$$

$$\Rightarrow^1 (AAI, q, b^2) \quad \text{by } A, i \xrightarrow{b} \varepsilon, q$$

$$\Rightarrow^1 (AI, q, b) \quad \text{by } A, q \xrightarrow{b} \varepsilon, q$$

$$\Rightarrow^1 (I, q, \varepsilon) \quad \text{by } A, q \xrightarrow{b} \varepsilon, q$$

$$(I, i, a^3b^3) \Rightarrow^1 (AI, i, a^2b^3) \quad \text{by } I, i \xrightarrow{a} AI, i$$

$$\Rightarrow^1 (AAI, i, ab^3) \quad \text{by } A, i \xrightarrow{a} AA, i$$

$$\Rightarrow^1 (AAAI, i, b^3) \quad \text{by } A, i \xrightarrow{a} AA, i$$

$$\Rightarrow^1 (AAI, q, b^2) \quad \text{by } A, i \xrightarrow{b} \varepsilon, q$$

$$\Rightarrow^1 (AI, q, b) \quad \text{by } A, q \xrightarrow{b} \varepsilon, q$$

$$\Rightarrow^1 (I, q, \varepsilon) \quad \text{by } A, q \xrightarrow{b} \varepsilon, q$$

$$\Rightarrow^1 (\varepsilon, f, \varepsilon) \quad \text{by } I, q \xrightarrow{\varepsilon} \varepsilon, f$$

$L(M)$, language accepted by a NPDA M

If $M = (Q, \Sigma, s, F, \Gamma, I, \Delta)$, then

$$L(M) = \{w \in \Sigma^* \mid (I, i, w) \Rightarrow^* (S, q, \varepsilon) \text{ holds for some } S \in \Gamma^* \text{ and } q \in F\}.$$

Example NPDA

States: $i \quad q \quad f$

Input symbols: $a \quad b$

Start state: i

Accepting state: f

Stack symbols: $I \quad A$

Initial stack symbol: I

In this case
$L(M) = \{a^n b^n \mid n \geq 1\}$

Transitions:

ε-transition	a-transitions	b-transitions
$I, q \xrightarrow{\varepsilon} \varepsilon, f$	$I, i \xrightarrow{a} AI, i$	$A, i \xrightarrow{b} \varepsilon, q$
	$A, i \xrightarrow{a} AA, i$	$A, q \xrightarrow{b} \varepsilon, q$

Theorem

A language is context-free if and only if it is accepted by some push-down automaton.

For a proof, see for example Hopcroft and Ullman Sect. 5.5.

Theorem

A language is context-free if and only if it is accepted by some push-down automaton.

For a proof, see for example Hopcroft and Ullman Sect. 5.5.

NB • $NFA^{\varepsilon} \subseteq NPDA$

Theorem

A language is context-free if and only if it is accepted by some push-down automaton.

For a proof, see for example Hopcroft and Ullman Sect. 5.5.

NB

- NFA $\subseteq$ NPDA
- acceptance "by empty stack"

Theorem

A language is context-free if and only if it is accepted by some non-deterministic push-down automaton.

For a proof, see for example Hopcroft and Ullman Sect. 5.5.

NB

- NFA $\subseteq$ NPDA
- acceptance "by empty stack"
- deterministic PDA less powerful

The way ahead, in Theory

- What does it mean for a function to be computable
  [IB Computation Theory]
- Are some computational tasks intrinsically unfeasible?
  [IB Complexity Theory]
- How can we rigorously specify & reason about the behaviour of programs?
  [IB Semantics of PLs , IB Logic & Proof]