Prove the following languages are context-free by designing context-free grammars which generate them:

$L_1 = \{a^n b^n c^p \ : \ n, p \geq 0\}$

$L_2 = \{a^n b^p c^n \ : \ n, p \geq 0\}$

$L_3 = \{a^n b^m : n \neq m, \ n, m \geq 0\}$

Hint: $L_3 = \{a^n b^m : n < m\} \cup \{a^n b^m : n > m\}$

$L_4 = \{ c u c v c : |u| = |v|, \ u, v \in \{a, b\}^* \}$

What is $L_1 \cap L_2$?

Announcements:

Assignment 3 has been posted.
If you do the 10 point bonus question, make sure you submit your code on connex.

Midterm: Wednesday June 27, in class.
Midterm tutorial?

Old midterms are available from the course web page.

Assignment #1 and #2 solutions are available in the connex resources.

Pushdown Automata

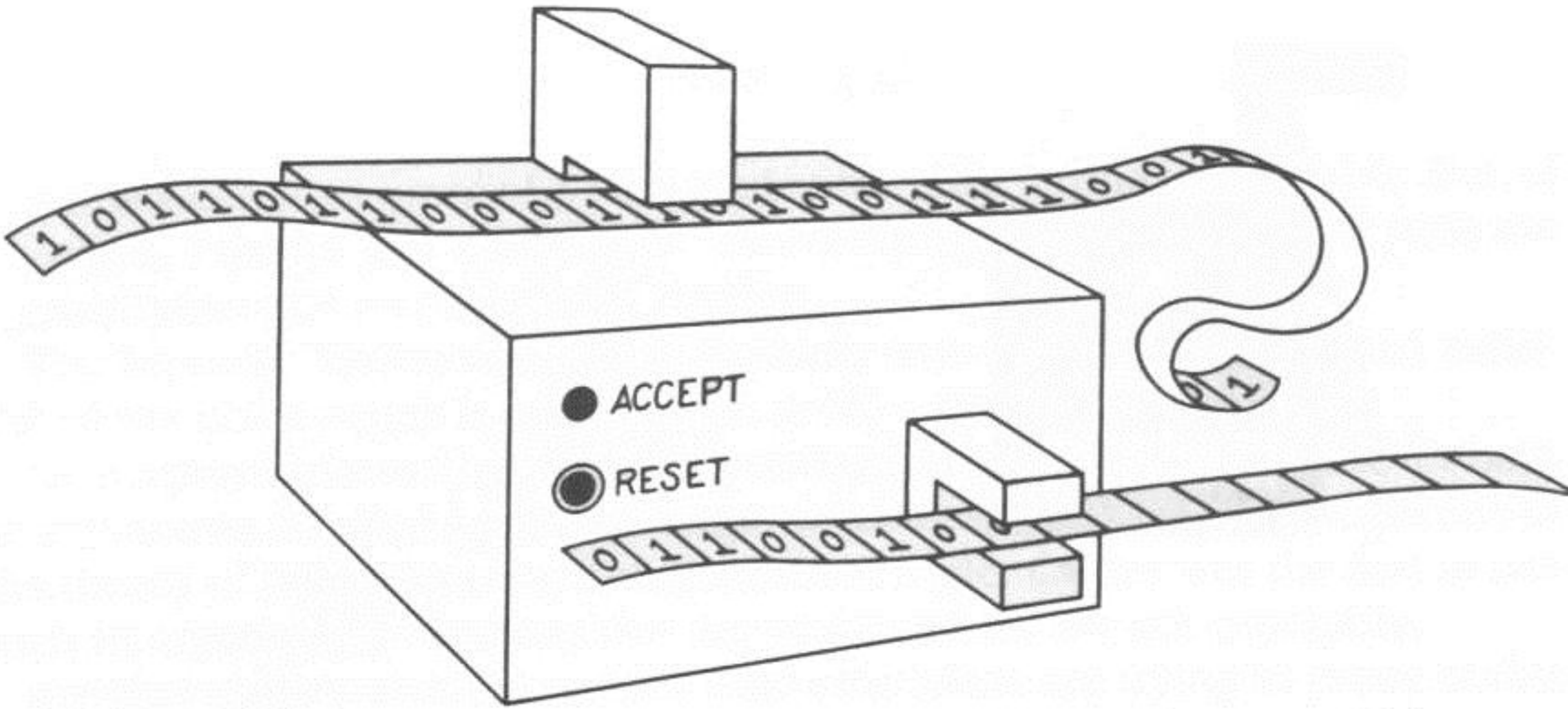


Figure 7.4 A push-down automaton

Picture from: Torsten Schaßan

Pushdown Automata:

A pushdown automaton is like a NDFA which has a stack.

Every context-free language has a pushdown automaton that accepts it.

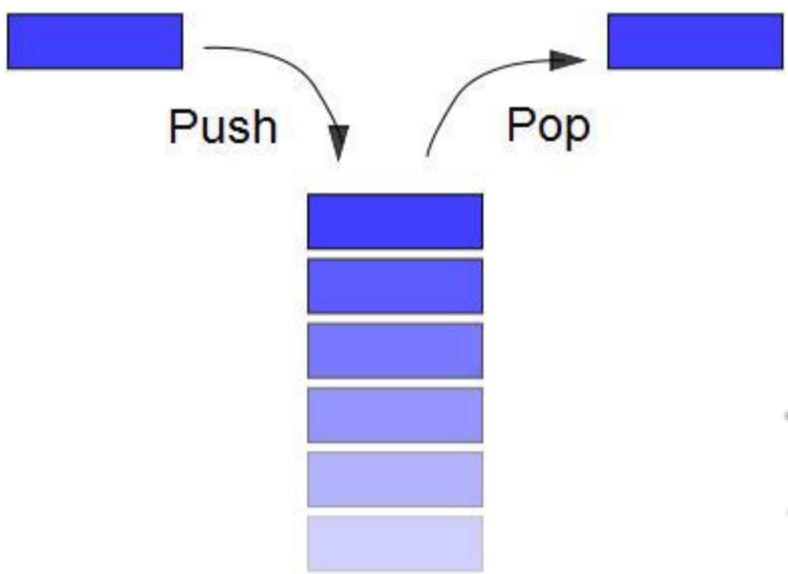
This lecture starts with some examples, gives the formal definition, then investigates PDA's further.

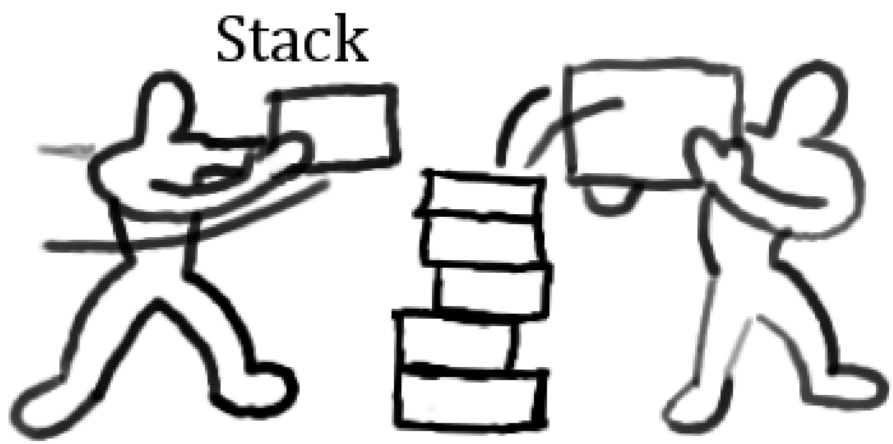
Stacks


corbis


SO LONG, AND THANKS FOR ALL THE FISH
DOUGLAS ADAMS
Thoreau
WALDEN
BARNES & NOBLE
The Gormenghast Novels
TITUS GROAN
GORMENGHAST
TITUS ALONE
Mervyn Peake
Potts
Sable
Smith
TEXTPATTERN SOLUTIONS
Bulletproof Web Design
THE PROUD HIGHWAY
HUNTER S. THOMPSON

Stack Data Structure: permits push and pop at the top of the stack.





$L = \{ w\, c\, w^R : w \in \{a, b\}^* \}$

Start state: s, Final State: {t}

State	Input	Pop	Next state	Push
s	a	ε	s	A
s	b	ε	s	B
s	c	ε	t	ε
t	a	A	t	ε
t	b	B	t	ε

To accept, there must exist a computation which:

1. Starts in the start state with an empty stack.
2. Applies transitions of the PDA which are applicable at each step.
3. Consumes all the input.
4. Terminates in a final state with an empty stack.

$w = a\ b\ b\ c\ b\ b\ a$

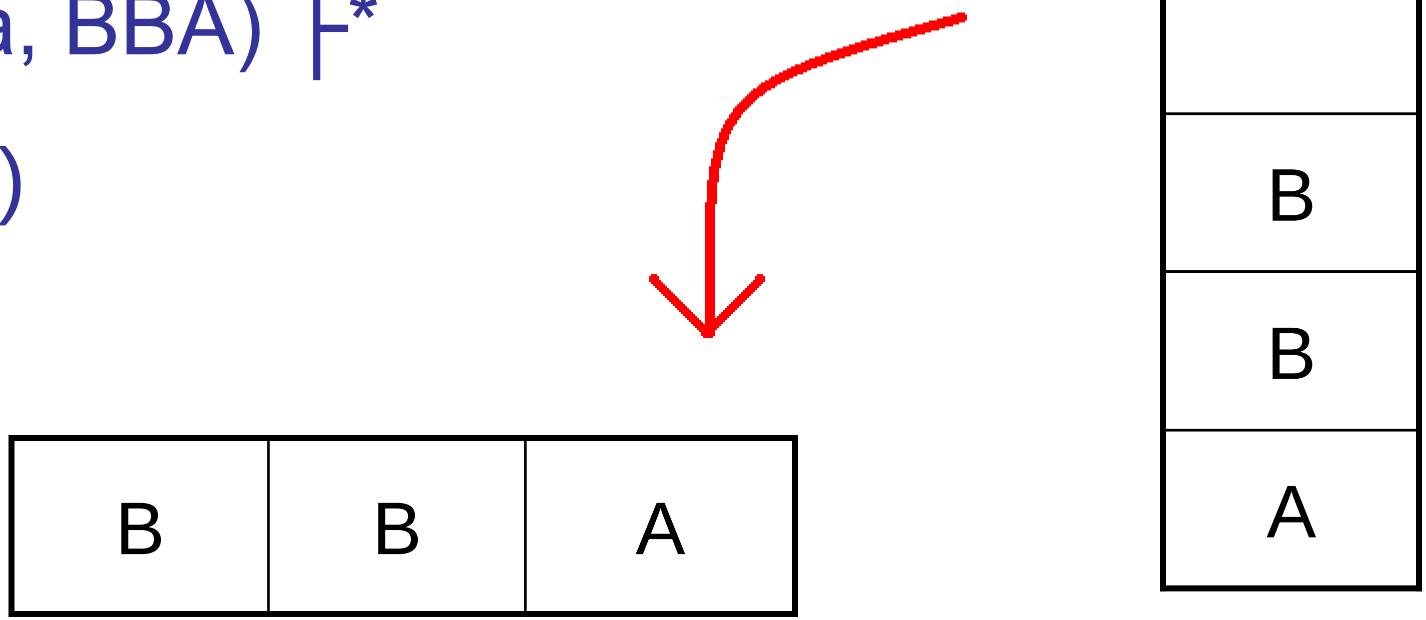
$(s, abbcbba, \varepsilon) \vdash^*$

$(s, cbba, BBA) \vdash$

$(t, bba, BBA) \vdash^*$

$(t, \varepsilon, \varepsilon)$

Stack is knocked over like this:



A pushdown automaton is a sextuple

$M = (K, \Sigma, \Gamma, \Delta, s, F)$ where

K is a finite set of states,

Σ is an alphabet (the input symbols)

Γ is an alphabet (the stack symbols)

Δ, the transition relation,

is a finite subset of

$$(\underbrace{K}_{\text{state}} \times \underbrace{(\Sigma \cup \{\varepsilon\})}_{\text{input}} \times \underbrace{\Gamma^*}_{\text{pop}}) \times (\underbrace{K}_{\text{next state}} \times \underbrace{\Gamma^*}_{\text{push}})$$

A configuration of a PDA is a member of

$K \quad \times \quad \Sigma^* \quad \times \quad \Gamma^*$

current state input remaining stack

A configuration $(q, \sigma w, \alpha x) \vdash (r, w, \beta x)$ if $((q, \sigma, \alpha), (r, \beta)) \in \Delta$.

For $M = (K, \Sigma, \Gamma, \Delta, s, F)$,

$L(M)$ (the language accepted by M) =

$\{ w \in \Sigma^* : (s, w, \varepsilon) \vdash^* (f, \varepsilon, \varepsilon)$ for some final state f in $F\}$.

$L(M) = \{ w \in \Sigma^* : (s, w, \varepsilon) \vdash^* (f, \varepsilon, \varepsilon)$ for some final state f in F$\}$.

To accept, there must exist a computation which:

1. Starts in the start state with an empty stack.
2. Applies transitions of the PDA which are applicable at each step.
3. Consumes all the input.
4. Terminates in a final state with an empty stack.

PDA's are non-deterministic:

$L = \{ w w^R : w \in \{a, b\}^* \}$

Start state: s, Final State: {t}

State	Input	Pop	Next state	Push
s	a	ε	s	A
s	b	ε	s	B
s	ε	ε	t	ε
t	a	A	t	ε
t	b	B	t	ε

Guessing wrong time to switch from s to t gives non-accepting computations.

L= {w in {a, b}* : w has the same number of a's and b's}

Start state: s

Final states: {s}

State	Input	Pop	Next state	Push
s	a	ε	s	B
s	a	A	s	ε
s	b	ε	s	A
s	b	B	s	ε

L= {w in {a, b}* : w has the same number of a's and b's}

State state: s, Final states: {f}

State	Input	Pop	Next state	Push
s	ε	ε	t	X
t	a	X	t	BX
t	a	A	t	ε
t	a	B	t	BB
t	b	X	t	AX
t	b	A	t	AA
t	b	B	t	ε
t	ε	X	f	ε

A more deterministic solution:

Stack will never contain both A's and B's.

X- bottom of stack marker.

1. Draw a parse tree for the following derivation:

$$S \Rightarrow C\ A\ C \Rightarrow C\ A\ b\ b \Rightarrow b\ b\ A\ b\ b \Rightarrow$$

$$b\ b\ B\ b\ b \Rightarrow b\ b\ a\ A\ a\ a\ b\ b$$

$$\Rightarrow b\ b\ a\ b\ a\ a\ b\ b$$

2. Show on your parse tree u, v, x, y, z as per the pumping theorem.

3. Prove that the language for this question is an infinite language.