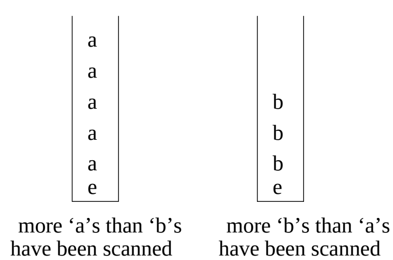
Another problem:

L = $\{\omega \in \{a, b\}^* \mid \omega$ contains an equal number of ‘a’s and ‘b’s$\}$

We use the stack to hold excess ‘a’s or ‘b’s.

We use a special stack symbol to indicate that an equal number of ‘a’s and ‘b’s have been scanned. ‘e’ denotes an empty stack.



$$
\begin{aligned}
\mathrm{M} &= (\mathrm{K}, \Sigma, \mathrm{T}, \Delta, q_0, \mathrm{F}) \\
\mathrm{K} &= \{\mathrm{s, q, f}\} \\
\Sigma &= \{\mathrm{a, b}\} \\
\mathrm{T} &= \{\mathrm{a, b, e}\} \\
q_0 &= \{\mathrm{s}\} \\
\mathrm{F} &= \{\mathrm{f}\}
\end{aligned}
$$

Slide Lecture 11 –321

$$
\begin{aligned}
\Delta = \{&((S,\lambda,\lambda),(q,e)) && (1)\\
&((q,a,e),(q,ae)) && (2)\\
&((q,a,a),(q,aa)) && (3)\\
&((q,a,b),(q,\lambda)) && (4)\\
&((q,b,e),(q,be)) && (5)\\
&((q,b,b),(q,bb)) && (6)\\
&((q,b,a),(q,\lambda)) && (7)\\
&((q,\lambda,e),(f,\lambda))\} && (8)
\end{aligned}
$$

Notes: (1) initializes the stack; ‘f’ indicates empty store has occurred.

Slide Lecture 11 –322

Operation of M on input abbbabaa

State	Unread Input	Stack Contents	Transition Used
s	abbbabaa	λ	-
q	abbbabaa	e	1
q	bbbabaa	ae	2
q	bbabaa	e	7
q	babaa	be	5

q	abaa	bbe	6
q	baa	be	4
q	aa	bbe	6
q	a	be	4
q	λ	e	4
f	λ	λ	8

NOTE: Sometimes PDAs are defined to include
an initial stack symbol.
For example, in Martin:

M	=	(Q, Σ, T, q_0, Z_0, Δ, F)
Q	=	a finite set of states
Σ	=	the input alphabet
T	=	the stack alphabet (Terminals and Nonterminals)
q_0	=	the initial state
Z_0	=	the initial stack symbol
Δ	=	is the transition function
F	=	set of final states

PDAs and Context-Free Grammars

PDAs are equivalent to context-free grammars
in the sense that they recognize
exactly the same class of languages.

In formal language theory a string ω
is generated by a context-free grammar
IFF there is a leftmost (rightmost) derivation
of ω from that grammar.

$\omega \in$ L(G) IFF

$$\omega_0 \overset{L}{\underset{G}{\Longrightarrow}} \omega_1 \overset{L}{\underset{G}{\Longrightarrow}} \omega_2 \ldots \overset{L}{\underset{G}{\Longrightarrow}} \omega_m$$

or simply

$$\omega_0 \overset{*L}{\underset{G}{\Longrightarrow}} \omega_m$$

Note: $\overset{L}{\underset{G}{\Longrightarrow}}$ means "yields a leftmost derivation".

PDA Lemma 1

For any context-free grammar G = (V, Σ, R, S),
A ∈ V, and $\omega \in \Sigma^*$

$$A \overset{*}{\underset{G}{\Longrightarrow}} \omega \quad \text{IFF} \quad A \overset{*L}{\underset{G}{\Longrightarrow}} \omega$$

Consider

$$A \overset{\Longrightarrow}{G} \alpha_1\beta_1\alpha_2\beta_2\cdots\alpha_k$$

$$\alpha_i \in \Sigma^*$$

$$\beta_i \in V$$

also define $b_i \in \Sigma^*$

In the leftmost derivation:

$$A \overset{\Longrightarrow}{G} \alpha_1 b_1\alpha_2\beta_2\cdots\alpha_k$$

but since the grammar is context-free,
we can derive $\beta_i \to b_i$ in any order for i.

We can therefore say "leftmost = rightmost."

PDA Lemma 2

If M is a PDA then

$$(q_1, \omega_1, \alpha_1) \overset{\overset{*}{\Longrightarrow}}{M} (q_2, \lambda, \alpha_2)$$

and

$$(q_2, \omega_2, \alpha_2\alpha_3) \overset{\overset{*}{\Longrightarrow}}{M} (q_3, \lambda, \alpha_4)$$

implies

$$(q_1, \omega_1\omega_2, \alpha_1\alpha_3) \overset{\overset{*}{\Longrightarrow}}{M} (q_3, \lambda, \alpha_4)$$

(this simply expresses the transitive nature of $\overset{\overset{*}{\Longrightarrow}}{M}$).

Furthermore

$$(q_1, \omega_1, \alpha_1) \overset{*}{\underset{M}{\Longrightarrow}} (q_2, \lambda, \lambda)$$

and

$$(q_2, \omega_2, \alpha_3) \overset{*}{\underset{M}{\Longrightarrow}} (q_3, \lambda, \lambda)$$

implies

$$(q_1, \omega_1\omega_2, \alpha_1\alpha_3) \overset{*}{\underset{M}{\Longrightarrow}} (q_3, \lambda, \lambda)$$

(here we have substiuted $\alpha_2 = \alpha_4 = \lambda$).

Lemma 2 expresses the idea that the PDA need scan no input except under the tape head, and also need scan only the top of the stack.

This does not answer how the PDA "chooses" the right move when the PDA is nondeterministic.

PDA Lemma 3

Every context-free grammar represents a language recognized by some PDA.
Let G = (V, Σ, R, S) be a context-free grammar.
We can construct a PDA M such that

$$L(G) = L(M)$$

We construct a two state PDA which begins in state p, moves to state q in the first step, and remains in q for the duration of the computation.

Slide Lecture 11 –333

$$\begin{aligned} M &= (K, \Sigma, T, \Delta, q_0, F) \\ K &= \{p, q\} \\ \Sigma &= \Sigma \\ T &= V \cup \Sigma \\ q_0 &= \{p\} \\ F &= \{q\} \end{aligned}$$

Slide Lecture 11 –334

The transitions are built as follows:

$$\Delta = \{((p, \lambda, \lambda), (q, S))$$

For each $A \rightarrow x \in R$, $((q, \lambda, A), (q, x))$,

For each $a \in \Sigma$, $((q, a, a), (q, \lambda))\ \}$

M begins by pushing S (the start symbol) on the stack and entering state q.

Operations:

1. Replace the nonternimal at the top of the stack by the righthand side of one of its rewrite rules
2. Pop a terminal off the stack, **provided** it matches the input symbol scanned.

Example:

G = (V, Σ, R, S)

V = {S}, Σ = {a, b, c} and

$$
\begin{aligned}
R \quad = \quad & \{S \longrightarrow aSa, \\
& S \longrightarrow bSb, \\
& S \longrightarrow c\}
\end{aligned}
$$

which generates $\{\omega c \omega^R \mid \omega \in \{a, b\}^*\}$

Slide Lecture 11 –337

By construction

M = ({p, q}, T, Σ, Δ, q_0 = {p}, F = {q}) where

$$
\begin{aligned}
\Delta \quad = \quad & \{((p, \lambda, \lambda), (q, S)), && (1) \\
& ((q, \lambda, S), (q, aSa)), && (2) \\
& ((q, \lambda, S), (q, bSb)), && (3) \\
& ((q, \lambda, S), (q, c)), && (4) \\
& ((q, a, a), (q, \lambda)), && (5) \\
& ((q, b, b), (q, \lambda)), && (6) \\
& ((q, c, c), (q, \lambda))\} && (7)
\end{aligned}
$$

Slide Lecture 11 –338

Derivation of abbcbba

State	Unread Input	Stack	Transition Used
p	abbcbba	λ	-
q	abbcbba	S	1
q	bbcbba	aSa	2
q	bbcbba	Sa	5
q	bcbba	bSba	3

q	bcbba	Sba	6
q	bcbba	bSbba	3
q	cbba	Sbba	6
q	cbba	cbba	4
q	bba	bba	7
q	ba	ba	6
q	a	a	6
q	λ	λ	5

To prove L(M) = L(G) we prove

1. If S $\overset{*L}{\underset{G}{\Longrightarrow}}$ $\alpha_1\alpha_2$,

   where $\alpha_1 \in \Sigma^*$ and $\alpha_2 \in V((V \cup \Sigma)^* \cup \lambda)$,

   then $(q, \alpha_1, S) \overset{*}{\underset{M}{\vdash}} (q, \lambda, \alpha_2)$

2. If $(q, \alpha_1, S) \overset{*}{\underset{M}{\vdash}} (q, \lambda, \alpha_2)$

   where $\alpha_1 \in \Sigma^*$ and $\alpha_2 \in (V \cup \Sigma)^*$,

   then S $\overset{*L}{\underset{G}{\Longrightarrow}}$ $\alpha_1\alpha_2$

Slide Lecture 11 –341

Collectively, these two claims verify the Lemma 3,
since (taking $\alpha_2 = \lambda$),

S $\overset{*L}{\underset{G}{\Longrightarrow}}$ α IFF $(q, \alpha, S) \overset{*}{\underset{M}{\vdash}} (q, \lambda, \lambda)$, $\alpha \in \Sigma^*$

Slide Lecture 11 –342

Claim 1, by Induction:

Assume S $\overset{*L}{\underset{G}{\Longrightarrow}}$ $\alpha_1\alpha_2$.

Basis: A zero length derivation. In this case S = α, $\alpha_1 = \lambda, \alpha_2 = S$.

Thus

$$(q, \alpha_1, S) \overset{*}{\underset{M}{\vdash}} (q, \lambda, \alpha_2)$$

Slide Lecture 11 –343

Inductive Hypothesis:

Assume that if S $\overset{*L}{\underset{G}{\Longrightarrow}}$ $\alpha_1\alpha_2$
by a derivation of at most N steps,

then $(q, \alpha_1, S) \overset{*}{\underset{M}{\vdash}} (q, \lambda, \alpha_2)$.

Slide Lecture 11 –344

Inductive Step:

Let

$$S = \mu_0 \overset{*L}{\underset{G}{\Longrightarrow}} \mu_1 \overset{*L}{\underset{G}{\Longrightarrow}} \dots \overset{*L}{\underset{G}{\Longrightarrow}} \mu_{N+1} = \alpha$$

be a leftmost derivation with $\alpha = \alpha_1 \alpha_2$.

Since μ_N has at least one nonterminal, let

$$\mu_n = \beta_1 A \beta_2, \quad \mu_{N+1} = \beta_1 \gamma \beta_2$$

where $\beta_1 \in \Sigma^*, A \in V, \beta_2 \in (V \cup \Sigma)^*$

and $A \longrightarrow \gamma$.

Note this implies $\alpha = \beta_1, \gamma, \beta_2 = \alpha_1 \alpha_2$.

We can rewrite α_1 as $\beta_1 \delta$
such that $\delta \alpha_2 = \gamma \beta_2$.
Note that $\delta \in \Sigma^*$.

So $(q, \alpha_1, S) = (q, \beta_1 \delta, S)$

Therefore

$$(q, \alpha_1, S) \overset{*}{\underset{M}{\vdash}} (q, \beta_1\delta, S)$$

By the inductive hypothesis

$$((q, \beta_1, S) \overset{*}{\underset{M}{\vdash}} (q, \lambda, A\beta_2)$$

therefore:

$$(q, \beta_1\delta, S) \overset{*}{\underset{M}{\vdash}} (q, \delta, A\beta_2)$$

Slide Lecture 11 –347

Given that our machine can process $((q, \lambda, A)(a, \gamma))$
the rule $(A \longrightarrow \gamma)$ implies a transition

$$((q, \lambda, A\beta_2)(a, \gamma\beta_2))$$

Therefore

$$(q, \delta, A\beta_2) \overset{*}{\underset{M}{\vdash}} (q, \delta, \gamma\beta_2)$$

and since we can rewrite α_1 as $\beta_1\delta$ such that $\delta\alpha_2 = \gamma\beta_2$.

$$(q, \delta, \gamma\beta_2) \overset{*}{\underset{M}{\vdash}} (q, \lambda, \alpha_2)$$

And the induction is complete.

Slide Lecture 11 –348

Claim 2 reverses this process and has a similar proof.