An Alternative Method of Language Acceptance by PDAs
Another method for associating a language with a PDA is to consider the language that consists of those input words for which there is a computation that leads to the emptying of the pushdown store. This is captured by the following definition.

**Definition**

Let \( M = (A, Z, Q, \delta, q_0, z_0, F) \) be a PDA. The **language accepted by \( M \) with an empty store** is given by

\[
N(M) = \{ x \in A^* \mid (z_0, q_0, x) \xrightarrow{\star}_M (\lambda, q, \lambda) \text{ for some } q \in Q \}.
\]

The set \( F \) plays no role in the definition of \( N(M) \).
Theorem

For every pda $M$ there is a pda $M'$ such that $L(M) = N(M')$. 
Proof

Let $M = (A, Z, Q, \delta, q_0, z_0, F)$. We have

$$L(M) = \{ x \in A^* \mid (z_0, q_0, x) \xrightarrow{*} (w, q, \lambda) \text{ for some } w \in Z^* \text{ and } q \in F \}.$$ 

Define the pda $M' = (A, Z \cup \{z'\}, Q \cup \{q', q'_0\}, \delta', q'_0, z', \emptyset)$, where $q', q'_0$ are two new states, and $z'$ is a new initial pushdown symbol, where $z' \notin Z$. The transition function $\delta'$ is given by

$$\delta'(z, q, a) = \begin{cases} 
\{(z'z_0, q_0)\} & \text{if } (z, q, a) = (z', q'_0, \lambda), \\
\delta(z, q, a) & \text{if } q \in Q - F, a \in A \cup \{\lambda\}, z \in Z, \\
\delta(z, q, a) \cup \{(\lambda, q')\} & \text{if } q \in F, z \in Z \cup \{z'\}, a \in A \cup \{\lambda\}, \\
\{(\lambda, q')\} & \text{if } q = q', z \in Z \cup \{z'\}, a = \lambda, \\
\emptyset & \text{in any other case.}
\end{cases}$$
The symbol $z'$ was introduced for the pda $M'$ since some words in 
$A^* - L(M)$ may empty the pushdown store of $M$. The presence of $z'$ at 
the bottom of the pushdown store makes this impossible in $M'$.

Since $\delta(z', q'_0, \lambda) = \{(z'z_0, q_0)\}$, $M'$ begins its work by entering the state
$q_0$ and by placing $z_0$ at the top of the pushdown store. If $x \in L(M)$, then
$(z_0, q_0, x) \xymatrix{\vdash\ar@{-}[r]_{M} & (w, q, \lambda)}$ for some $w \in Z^*$ and $q \in F$. Correspondingly, in
$M'$ we have

$\quad (z', q'_0, x) \xymatrix{\vdash\ar@{-}[r]_{M'} & (z'z_0, q_0, x) \xymatrix{\vdash\ar@{-}[r]_{M'} & (z'w, q, \lambda) \xymatrix{\vdash\ar@{-}[r]_{M'} & (\lambda, q', \lambda),}$

by the definition of $\delta'$. This implies $L(M) \subseteq N(M')$. 
To prove the converse inclusion, let $x \in N(M')$, so $(z', q'_0, x) \vdash_{M'}^* (\lambda, q, \lambda)$ for some state $q \in Q \cup \{q', q'_0\}$. The definition of $\delta'$ implies that this computation necessarily has the form

$$(z', q'_0, x) \vdash_{M'} (z'z_0, q_0, x) \vdash_{M'}^* (\lambda, q, \lambda),$$

since there exists only one transition for the triple $(z', q', \lambda)$, namely $(z'z_0, q_0)$. Note that the symbol $z'$ can be erased by $M'$ only if this pda reaches a state $q \in F \cup \{q'\}$. Let $u$ be the suffix of $x$ that remains to be read when $M'$ reached the state $q'$ for the first time. Since $M'$ enters $q'$ only from a final state $q_1$ of $M$ we have:

$$(z', q'_0, x) \vdash_{M'} (z'z_0, q_0, x) \vdash_{M'}^* (w, q_1, u) \vdash_{M'} (w', q', u) \vdash_{M'}^* (\lambda, q, \lambda),$$
Once $M'$ enters the state $q'$ no symbol is read from the input, so we have $u = \lambda$. This allows us to write the previous computation as

$$(z', q'_0, x) \vdash_{M'} (z'z_0, q_0, x) \vdash^*_{M'} (w, q_1, \lambda) \vdash_{M'} (w', q', \lambda) \vdash^*_{M'} (\lambda, q, \lambda),$$

and this implies the existence of the computation

$$(z_0, q_0, x) \vdash^*_{M} (w, q_1, \lambda),$$

which, in turn, implies $x \in L(M)$. This proves the needed inclusion $N(M') \subseteq L(M)$. 

(Proof cont’d)
Theorem

For every context-free grammar $G$ there is a one-state PDA $M$ such that $L = N(M)$. 
Suppose that \( L = L(G) \), where \( G = (A_N, A_T, S, P) \) is a context-free grammar. Let \( M = (A_T, A_N \cup A_T, \{q_0\}, \delta, q_0, S, \emptyset) \) be a pda whose transition function is given by

\[
\delta(X, q_0, \lambda) = \{ (\alpha^R, q_0) \mid X \rightarrow \alpha \in P \},
\]
\[
\delta(a, q_0, a) = \{ (\lambda, q_0) \},
\]

for every \( a \in A_T \), \( X \in A_N \), and \( \delta(s, q_0, a) = \emptyset \) in all other cases. Let

\[
S = \gamma_0 \xrightarrow{G} \gamma_1 \xrightarrow{G} \cdots \xrightarrow{G} \gamma_n = u\alpha
\]

be a leftmost derivation of \( u\alpha \) in \( G \), where \( u \in A_T^* \) and \( \alpha \in (A_N \cup A_T)^* \) is either the null word or a word that begins with a nonterminal symbol. We claim that \((S, q_0, uw) \vdash^* (\alpha^R, q_0, w)\) for every \( w \in A_T^* \). The argument is by induction on \( n \).
For $n = 0$, we have $u = \lambda$ and $\alpha = S$. Thus, the claim is simply

$$(S, q_0, w) \vdash^* \mathcal{M} (S, q_0, w),$$

which follows from the definition of $\vdash^* \mathcal{M}$.

For the induction step suppose that

$$S = \gamma_0 \xrightarrow{G} \cdots \xrightarrow{G} \gamma_n \xrightarrow{G} \gamma_{n+1} = u\alpha$$

is a leftmost derivation, where $\gamma_n = u'X\theta$ and $\gamma_{n+1} = u'u''\beta\theta$. In other words, the last step of the derivation uses the production $X \rightarrow u''\beta$, where $u'' \in A_T^*$ and $\beta \in (A_N \cup A_T)^*$ is either the null word or a word that begins with a nonterminal symbol.
Thus, the derivation above may be written

$$S \xrightarrow[n]{G} u'X\theta \xrightarrow[G]{u''\beta\theta},$$

and we have the following computation of $M$:

$$(S, q_0, u'u''w) \vdash^* ((\theta R^X, q_0, u''w) = (\theta^R X, u''w) \quad \text{(by the inductive hypothesis)}$$

$$\vdash (\theta^R \beta^R u''^R, q_0, u''w) \quad \text{(since } (\beta^R u''^R, q_0) \in \delta(X, q_0, \lambda))$$

$$\vdash^* (\theta^R \beta^R, q_0, w).$$

The last line follows from the observation that $\delta(a, q_0, a) = \{(\lambda, q_0)\}$ for each $a \in A_T$ implies that $(x^R, q_0, x) \vdash^* (\lambda, q_0, \lambda)$ for every $x \in A_T^*$. Since $\theta^R \beta^R = (\beta\theta)^R$, we have completed the induction step. Therefore, if $u \in L(G)$ we have $S \xrightarrow[n]{G} u$, and this implies $(S, q_0, u) \vdash^* (\lambda, q_0, \lambda)$, which shows that $u \in N(M)$, hence $L(G) \subseteq N(M)$. 
To prove that $N(M) \subseteq L(G)$, we show that $(X, q_0, u) \xrightarrow{\ast}{M} (\lambda, q_0, \lambda)$ implies $X \xrightarrow{G} u$ for $X \in A_N$ and $u \in A_T^*$.

We factor the input word $u$ into a series of subwords $u = u_0u_1 \cdots u_{k-1}$, each corresponding to a certain change in the pushdown store. Specifically, the top symbol of the pushdown store of each step of the computation can be either a terminal or a nonterminal symbol. Any step at which a nonterminal is at the top determines the boundary between a $u_i$ and its successor $u_{i+1}$ in the input. Thus, $u_i$ could be empty (when a nonterminal at the top is replaced by a nonterminal) or could contain several symbols (when there are terminal symbols at the top that are popped off by transitions of the form $(\lambda, q_0) \in \delta(a, q_0, a)$).
Thus, we can write $u = u_0u_1 \cdots u_{k-1}$, where $u_i \in A_T^*$ for $0 \leq i \leq k - 1$, and

\[
(X, q_0, u) = (\gamma_0, q_0, u_0u_1 \cdots u_{k-1}) \\
\vdash^* (\gamma_1, q_0, u_1 \cdots u_{k-1}) \\
\vdots \\
\vdash^* (\gamma_{k-1}, q_0, u_{k-1}) \\
\vdash^* (\lambda, q_0, \lambda),
\]

where each $\gamma_i$ has the form $\gamma'_iX$ for $\gamma'_i \in (A_T \cup A_N)^*$ and $X \in A_N$. 
The definition of $\mathcal{M}$ implies that the computation

$$(\gamma_i, q_0, u_i \cdots u_{k-1}) \cdot^{*} \mathcal{M} (\gamma_{i+1}, q_0, u_{i+1} \cdots u_{k-1})$$

can be written as

$$(\gamma_i, q_0, u_i \cdots u_{k-1}) = (\gamma'_i X_{p_i}, q_0, u_i \cdots u_{k-1})$$

$$\vdash (\gamma'_i \alpha_{p_i}^R, q_0, u_i \cdots u_{k-1})$$

$$\vdash^* (\gamma_{i+1}, q_0, u_{i+1} \cdots u_{k-1}),$$

where $X_{p_i} \rightarrow \alpha_{p_i} = u_i \beta_{p_i}$ is a production of $G$ such that $\beta_{p_i} \in (A_N \cup A_T)^*$ is the null word or a word that begins with a nonterminal symbol, and $\gamma_{i+1} = \gamma'_i \beta_{p_i}^R$ for $0 \leq i \leq k - 1$.

We prove by induction on $\ell$ that we have the leftmost derivation

$$\gamma_{k-1-\ell}^R \Rightarrow^*_{G} u_{k-1-\ell} \cdots u_{k-1}.$$
An Alternative Method of Language Acceptance by PDAs

For $\ell = 0$ we have

$$\left( \gamma_{k-1}, q_0, u_{k-1} \right) \models M \left( u_{k-1}^R, q_0, u_{k-1} \right) \models M \left( \lambda, q_0, \lambda \right),$$

because $\gamma_{k-1}$ is the last content of the pushdown store that may contain a nonterminal, which means that $\gamma_{k-1} = X \in A_N$ and $X \rightarrow u_{k-1} \in P$. Therefore, $\gamma_{k-1}^R = X \Rightarrow^* u_{k-1}$. 
Suppose that $\gamma_{i+1}^R \Rightarrow_G u_{i+1} \cdots u_{k-1}$; that is, $\beta_p \gamma_{i}^R \Rightarrow_G u_{i+1} \cdots u_{k-1}$. This implies $u_{i} \beta_p \gamma_{i}^R \Rightarrow_G u_{i} u_{i+1} \cdots u_{k-1}$, so $\alpha_p \gamma_{i}^R \Rightarrow_G u_{i} u_{i+1} \cdots u_{k-1}$.

The existence of the production $X_i \rightarrow \alpha_p$ allows us to write

$$X_i \gamma_{i}^R \Rightarrow_G \alpha_p \gamma_{i}^R \Rightarrow_G u_{i} u_{i+1} \cdots u_{k-1},$$

and $X_i \gamma_{i}^R = (\gamma_{i} X_i)^R = \gamma_{i}^R$.

Choosing $X = S$ we conclude that $x \in N(M)$ implies $(S, q_0, u) \vdash_{M}^{*} (\lambda, q_0, \lambda)$, which in turn, implies $S \Rightarrow_G^{*} u$ and $u \in L(G)$. 

Note that a computation of the pda $M$ that leads to the acceptance of a word $u$ uniquely defines a leftmost derivation in the grammar $G$.

**Example**

Consider the nonambiguous context-free grammar

$$G_{ae} = (\{X_e, X_t, X_f\}, \{+,-,\ast,/, (, ), v, n\}, X_e, P)$$

introduced before which generates the language of parenthesized arithmetic expressions. The pda that accepts the language $L(G_{ae})$ with an empty pushdown store is

$$M = (\{+,-,\ast,/, (, ), v, n\}, \{X_e, X_t, X_f, +, -, \ast, /, (, ), v, n\},\{q_0\}, \delta, q_0, X_e, \emptyset),$$

where $\delta$ is specified by the table in the next slide.
(Example cont’d)

<table>
<thead>
<tr>
<th>Top</th>
<th>State</th>
<th>Input</th>
<th>Transition Function $\delta(z, q, a)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$z$</td>
<td>$q_0$</td>
<td>$\lambda$</td>
<td>${(X_t + X_e, q_0), (X_t - X_e, q_0), (X_t, q_0)}$</td>
</tr>
<tr>
<td>$X_e$</td>
<td>$q_0$</td>
<td>$\lambda$</td>
<td>${(X_f \ast X_t, q_0), (X_f / X_t, q_0), (X_f, q_0)}$</td>
</tr>
<tr>
<td>$X_t$</td>
<td>$q_0$</td>
<td>$\lambda$</td>
<td>${(v, q_0), (n, q_0), ()X_e(, q_0)}$</td>
</tr>
<tr>
<td>$X_f$</td>
<td>$q_0$</td>
<td>$a$</td>
<td>${()X_e, (\lambda, q_0)}$</td>
</tr>
</tbody>
</table>

The last line of the table applies to every symbol $a \in \{+, -, \ast, /, (,), v, n\}$. If $\delta(z, q_0, a)$ is not mentioned in the table, then $\delta(z, q_0, a) = \emptyset$. 
The word \((n + n) \ast n\) can be generated in \(G_{ae}\) using the leftmost derivation

\[
\begin{align*}
X_e & \Rightarrow X_t \quad \Rightarrow X_t \ast X_f \quad \Rightarrow X_f \ast X_f \\
& \Rightarrow (X_e) \ast X_f \quad \Rightarrow (X_e + X_t) \ast X_f \\
& \Rightarrow (X_t + X_t) \ast X_f \quad \Rightarrow (X_f + X_t) \ast X_f \quad \Rightarrow (n + X_t) \ast X_f \\
& \Rightarrow (n + X_f) \ast X_f \quad \Rightarrow (n + n) \ast X_f \quad \Rightarrow (n + n) \ast n.
\end{align*}
\]

that corresponds to the derivation tree shown next:
An Alternative Method of Language Acceptance by PDAs

\[
X_t \\
X_e \quad X_f \\
X_e \quad * \\
X_f \\
X_e \\
X_t \\
X_f \\
+ \\
X_t \\
X_f \\
X_e \\
X_t \\
X_f \\
n \\
n \\
n
\]
The computation that leads to the acceptance of the word \((n + n) \ast n\) in \(M\) is

\[
(X_e, q_0, (n + n) \ast n) \vdash (X_t, q_0, (n + n) \ast n) \\
\vdash (X_f \ast X_t, q_0, (n + n) \ast n) \vdash (X_f \ast X_f, q_0, (n + n) \ast n) \\
\vdash (X_f \ast X_e, q_0, (n + n) \ast n) \vdash (X_f \ast X_e, q_0, (n + n) \ast n) \\
\vdash (X_f \ast X_t + X_e, q_0, n + n) \ast n) \vdash (X_f \ast X_t + X_t, q_0, n + n) \ast n) \\
\vdash (X_f \ast X_t + X_f, q_0, n + n) \ast n) \vdash (X_f \ast X_t + n, q_0, n + n) \ast n) \\
\vdash (X_f \ast X_t + , q_0, +n) \ast n) \vdash (X_f \ast X_t, q_0, n) \ast n) \\
\vdash (X_f \ast X_f, q_0, n) \ast n) \vdash (X_f \ast n, q_0, n) \ast n) \vdash (X_f \ast n, q_0 ,) \ast n) \\
\vdash (X_f \ast , q_0, *n) \vdash (X_f, q_0, n) \vdash (n, q_0, n) \vdash (\lambda, q_0, \lambda)
\]
For every pda $M$ the language $N(M)$ is context-free.
We need the following technical result showing that whenever there is a pda that accepts a language with an empty store, then there is a way to construct a pda that accepts the same language both with an empty store and by entering a final accepting state.

**Theorem**

For every pda $M = (A, Z, Q, \delta, q_0, z_0, F)$ there exists a pda $M' = (A, Z', Q', \delta', q'_0, z'_0, \{q'\})$ such that $(z'_0, q'_0, x) \vdash^{*}_{M'} (\lambda, q_1, \lambda)$ implies $q_1 = q'$ and $N(M) = N(M') = L(M')$. 
Proof

Pick $q'_0, q' \notin Q$ and $z'_0 \notin Z$. Define the pda

\[ M' = (A, Z', Q', \delta', q'_0, z'_0, \{ q' \}) \]

as $Q' = Q \cup \{ q'_0, q' \}$, $Z' = Z \cup \{ z' \}$, $\delta'(z'_0, q'_0, \lambda) = \{ (z'_0 z_0, q_0) \}$, $\delta'(z'_0, q, \lambda) = \{ (\lambda, q') \}$ for every $q \in Q$, and $\delta'(z, q, a) = \delta(z, q, a)$ in every other case. In other words, $M'$ begins by putting a marker, $z'_0$, onto the pushdown store and then simulating $M$ until $M$ would have emptied its pushdown store. At this time $M'$ removes the marker, thus emptying its store, and goes into a final state.
(Proof cont’d)

Let $x \in N(M)$. We have $(z_0, q_0, x) \xRightarrow{\star}{\mathcal{M}} (\lambda, q, \lambda)$ for some $q \in Q$.

Therefore, in $M'$ we have the computation

$$(z'_0, q'_0, x) \xrightarrow{\mathcal{M}'} (z'_0 z_0, q_0, x) \xRightarrow{\star}{\mathcal{M}'} (z'_0, q, \lambda) \xrightarrow{\mathcal{M}'} (\lambda, q', \lambda),$$

so $x \in N(M')$ and $x \in L(M')$, which shows that $N(M) \subseteq N(M')$ and $N(M) \subseteq L(M')$. 

(Proof cont’d)

Conversely, suppose that $x \in N(\mathcal{M}')$ or that $x \in L(\mathcal{M}')$.

In the first case, $(z'_0, q'_0, x) \vdash_{\mathcal{M}'} (\lambda, \bar{q}, \lambda)$ for some state $\bar{q} \in Q'$. The definition of $\mathcal{M}'$ implies that this computation can be written as

$$(z'_0, q'_0, x) \vdash_{\mathcal{M}'} (z'_0 z_0, q_0, x) \vdash_{\mathcal{M}'} (\lambda, \bar{q}, \lambda).$$

Note that in $\mathcal{M}'$ the symbol $z'_0$ cannot be erased unless $\mathcal{M}'$ switches to the state $q'$. Therefore, in the previous computation we have $\bar{q} = q'$, and this computation can be written as

$$(z'_0, q'_0, x) \vdash_{\mathcal{M}'} (z'_0 z_0, q_0, x) \vdash_{\mathcal{M}'} (z'_0, q, \lambda) \vdash_{\mathcal{M}'} (\lambda, q', \lambda)$$

for some $q \in Q$. Thus, we must have $(z_0, q_0, x) \vdash_{\mathcal{M}} (\lambda, q, \lambda)$, that is $x \in N(\mathcal{M})$. 

In the second case, $x \in L(M')$ implies $(z'_0, q'_0, x) \xrightarrow{\star}{M'} (w, q', \lambda)$. Observe that $M'$ may enter its final state $q'$ only by erasing the symbol $z'_0$ located at the bottom of the pushdown store. This implies that the above computation has the form

$$(z'_0, q'_0, x) \xrightarrow{\star}{M} (z'_0z_0, q_0, x) \xrightarrow{\star}{M'} (z'_0, q, \lambda) \xrightarrow{\star}{M'} (\lambda, q', \lambda).$$

As before, this implies the existence of the computation

$$(z_0, q_0, x) \xrightarrow{\star}{M} (\lambda, q, \lambda),$$

so $x \in N(M)$. We proved that $N(M') \subseteq N(M)$ and $L(M') \subseteq N(M)$. Thus, $N(M) = N(M') = L(M')$, which is the desired conclusion.
Theorem

If $L$ is a language such that $L = N(M)$ for some PDA $M$, then $L$ is a context-free language.
Proof

Suppose that $L = N(M)$, where $M = (A, Z, Q, \delta, q_0, F)$ is a pda. By Theorem 5 we can assume without loss of generality that $F = \{q_f\}$ and that $L = \{x \in A^* \mid (z_0, q_0, x)^* \vdash_M (\lambda, q_f, \lambda)\}$.

Consider the alphabet $\hat{Z} = \{z^{q_i q_j} \mid z \in Z, q_i, q_j \in Q\}$ and the context-free grammar $G = (\hat{Z}, A, z_0^{q_0 q_f}, P)$, whose set of productions $P$ is constructed as follows:
If \((z_{i_k} \cdots z_{i_0}, p) \in \delta(z, q, a)\), then place the following productions into \(P\):

\[z^{qq_{i_k}} \rightarrow az^{pq_{i_0}}z^{q_{i_0}q_{i_1}} \cdots z^{q_{i_k-1}q_{i_k}},\]

for every \(q_{i_0}, \cdots, q_{i_k} \in Q\).

If \((\lambda, p) \in \delta(z, q, a)\), then place the production \(z^{qp} \rightarrow a\) into \(P\).

Define the relation \(\rho \subseteq \hat{Z} \times \hat{Z}\) by \((z^{q_iq_j}, z^{q_kq_h}) \in \rho\) if and only if \(q_j = q_k\) and consider the regular language \(H = L_\rho\). Let \(d : \hat{Z}^* \rightarrow Z^*\) be the morphism defined by \(d(z^{q_iq_j}) = z\) for every \(z^{q_iq_j} \in \hat{Z}\).
(Proof cont’d)

We prove that for \( n \geq 1 \), we have the leftmost derivation \( z^{q_i q_j} \xrightarrow{n} w \alpha \) (where \( w \in A^* \) and \( \alpha \in \hat{Z}^* \)) if and only if \((z, q_i, wy) \xrightarrow{n} (d(\alpha)^R, p, y) \) and one of the following conditions is satisfied:

- \( \alpha \in H \), the first symbol of \( \alpha \) has the form \( z^{pq} \), and the last symbol of \( \alpha \) has the form \( z^{qq_j} \), or
- \( \alpha = \lambda \) and \( p = q_j \).
The argument is by induction on $n$. For the basis step, $n = 1$, suppose that $z^{q_i q_j} \Rightarrow_G w\alpha$. The production applied for this one-step derivation is either $z^{q_i q_j} \rightarrow a z_{i_0}^{p q_i_0} z_{i_1}^{q_{i_0} q_{i_1}} \cdots z_{i_k}^{q_{i_{k-1}} q_j}$ which implies

$$w = a \text{ and } \alpha = z_{i_0}^{p q_i_0} z_{i_1}^{q_{i_0} q_{i_1}} \cdots z_{i_k}^{q_{i_{k-1}} q_j},$$

or is $z^{q p} \rightarrow a$, which implies

$$w = a \text{ and } \alpha = \lambda,$$

respectively.
The first case may occur if and only if \((z_{i_k} \cdots z_{i_0}, p) \in \delta(z, q_i, a)\). Therefore, we have

\[
(z, q_i, ay) \vdash \mathcal{M} (z_{i_k} \cdots z_{i_0}, p, y) = (d(\alpha)^R, p, y)
\]

Also, the second case takes place if and only if \((\lambda, p) \in \delta(z, q_i, a)\) which is equivalent to

\[
(z, q_i, ay) \vdash \mathcal{M} (\lambda, p, y).
\]

This concludes the basis step.
For the inductive step assume that the statement holds for \( n \) and consider a leftmost derivation of length \( n + 1 \): 
\[
\begin{align*}
  z^{q_i q_j} & \xrightarrow{\text{G}} w' \alpha'.
\end{align*}
\]
Two cases may occur depending on form of the production applied in the last step of this derivation:

- If the production applied in the last step was
  
  \[
  z^{q_j q_{j_\ell}} \rightarrow a z^{j_0 z_{j_1} \cdots z_{j_{\ell-1}} q_{j_\ell}},
  \]
  
  then the derivation can be written as
  
  \[
  z^{q_i q_j} \xrightarrow{\text{G}} w z^{q_{j_\ell}} \alpha \xrightarrow{\text{G}} w a z^{j_0 z_{j_1} \cdots z_{j_{\ell-1}} q_{j_\ell}} \alpha.
  \]
  
  (1)
This takes place if and only if $w' = wa$, $\alpha' = z_{j_0}^{r_{q_{j_0}}} z_{j_1}^{q_{j_0}q_{j_1}} \cdots z_{j_{\ell-1}}^{q_{j_{\ell-1}}q_{j_\ell}} \alpha$. By the inductive hypothesis, the first part of the derivation takes place if and only if $z^{qq_{j_\ell}} \alpha \in H$ and

$$(z, q_i, way) \vdash^n_M (d(z^{qq_{j_\ell}} \alpha)^R, q, y) = (d(\alpha)^R z, q, ay).$$

The last step of the derivation can be executed if and only if

$$(z_{j_\ell} \cdots z_{j_0}, r) \in \delta(z, q, a),$$

by the definition of the grammar $G$. Thus, the derivation (1) takes place if and only if

$$(z, q_i, w'y) = (z, q_i, way) \vdash^n_M (d(\alpha)^R z, q, ay)$$

$$(d(\alpha)^R z_{j_\ell} \cdots z_{j_0}, r, y)$$

$$= (d(z_{j_0}^{r_{q_{j_0}}} z_{j_1}^{q_{j_0}q_{j_1}} \cdots z_{j_{\ell-1}}^{q_{j_{\ell-1}}q_{j_\ell}} \alpha)^R, r, y)$$

$$= (d(\alpha')^R, r, y).$$
If the production applied in the last step of the derivation was $z^{\ell} \rightarrow a$, the derivation can be written

$$z^{q_i q_j} \xrightarrow{n} w z^{q_i q_j} \alpha \xrightarrow{n} w a \alpha.$$  \hspace{1cm} (2)

Thus $w' = wa$ and $\alpha' = \alpha$. By the inductive hypothesis we have:

$$(z, q_i, w' y) = (z, q_i, w a y) \xrightarrow{n} \mathcal{M} (d(z^{q_i q_j} \alpha)^R, q, a y) = (d(\alpha)^R z, q, a y).$$

The existence of the production $z^{q_i q_j} \rightarrow a$ is equivalent to $(\lambda, q_j \ell) \in \delta(z, q, a)$, so the existence of the derivation (2) is equivalent to the existence of the computation

$$(z, q_0, w' y) \xrightarrow{n} \mathcal{M} (d(\alpha)^R z, q, a y) \xrightarrow{n} \mathcal{M} (d(\alpha)^R, q_j \ell, y) = (d(\alpha')^R, q_j \ell, y).$$

By taking $q_i = q_0$, $\alpha = \lambda$, $z = z_0$, $y = \lambda$, and $p = q_f$ in the initial claim, we conclude that a leftmost derivation $z^{q_0 q_f} \xrightarrow{n} w$ exists if and only if

$$(z_0, q_i, w) \xrightarrow{n} \mathcal{M} (\lambda, q_f, \lambda).$$

This shows that $L(G) = N(M)$, so $N(M)$ is indeed a context-free language.
Theorem

Let $L \subseteq A^*$ be a language over the alphabet $A$. The following statements are equivalent:

- There is a PDA $M$ such that $L = L(M)$.
- There is a PDA $M$ such that $L = N(M)$.
- There is a PDA $M$ (having a single final state) such that $L = N(M) = L(M)$.
- There is an one-state PDA $M$ such that $L = N(M)$.
- $L$ is a context-free language.