Lesson 4: Push-down automata

Theme: Push-down automata as a model of computation for context-free languages.

In the following we will have two alphabets $\Sigma$ and $\Gamma$. A push-down automaton (PDA) is a system $A = \langle \Sigma, \Gamma, Q, q_0, F, \delta \rangle$, where each of the component is as follows.

- $\Sigma$ is a finite alphabet, called the input alphabet, whose elements are called input symbols.
- $\Gamma$ is a finite alphabet, called the stack alphabet, whose elements are called stack symbols.
- $Q$ is a finite set of states.
- $q_0 \in Q$ is the initial state.
- $F \subseteq Q$ is the set of accepting states.
- $\delta \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times (\Gamma \cup \{\varepsilon\}) \times Q \times (\Gamma \cup \{\varepsilon\})$ is the transition relation.

We will usually write a transition $(p, x, y, q, z) \in \delta$ as:

$$(p, x, \text{pop}(y)) \rightarrow (q, \text{push}(z))$$

Intuitively, such transition means that when a PDA is in state $p$ reading $x$ from the input and the top of the stack is $y$, it can “pop” $y$ from the top of the stack and moves to state $q$ and push $z$ into the stack. Here it is possible that $x, y$ and $z$ are the empty string $\varepsilon$.

Note that the fashion a symbol is written into and taken out of the stack is “Last In First Out” (LIFO), i.e., the last symbol that gets written into the stack has to come out first. It is also important to note that while the input is a word over $\Sigma$, its stack contains symbols from $\Gamma$.

We will now describe formally how PDA computes. Let $A = \langle \Sigma, \Gamma, Q, q_0, F, \delta \rangle$ be a PDA. A configuration of $A$ is a pair $(q, u) \in Q \times \Gamma^*$, where $q$ is the state of $A$ and $u$ is the content of the stack. The initial configuration is $(q_0, \varepsilon)$. A configuration is accepting, if the state component is one of the accepting states.

On input $w = a_1 \ldots a_m$, a run of a PDA from a configuration $(q, u)$ is a sequence:

$$(p_0, v_0) \vdash_{b_1} (p_1, v_1) \vdash_{b_2} \cdots \vdash_{b_n} (p_n, v_n), \quad (\dagger)$$

where

- $(p_0, v_0) = (q, u),$
- $b_1 \cdots b_n = a_1 \cdots a_m$, i.e., some of the $b_i$’s can be $\varepsilon$,
- for each $i = 1, \ldots, n$, there is $(p_i, x, \text{pop}(y)) \rightarrow (p_{i+1}, \text{push}(z)) \in \delta$ such that
  - $x = b_i$,
  - $v_i = sy$ and $v_{i+1} = sz$, for some $s \in \Gamma^*$.

Note: Here $v_i = sy$ denotes that the content of the stack, where the top of the stack is $y$. When the transition $(p_i, x, \text{pop}(y)) \rightarrow (p_{i+1}, \text{push}(z))$ is applied, the PDA is in state $p_i$ reads $x$ from the input, “pops” $y$ from the stack, and moves to state $p_{i+1}$ and at the same time “pushes” $z$ into the stack. Thus, the subsequent content $v_{i+1}$ of the stack is $sz$. 

1/11
We will also write the run \[ (p_0, v_0) \vdash^* w (p_n, v_n) \].

In this case we will also say that there is a run of \( A \) on \( w \) from \( (p_0, v_0) \) to \( (p_n, v_n) \).

A run is accepting, if it starts from the initial configuration and ends with an accepting configuration. The language accepted by \( A \), denoted by \( L(A) \), consists of all the words for which it has an accepting run. Formally,

\[ L(A) = \{ w \mid \text{there is an accepting run of } A \text{ on } w \}. \]

The following theorem states that CFL and PDA are actually equivalent.

**Theorem 4.1**

- For every CFG \( G \), there is a PDA \( A \) such that \( L(A) = L(G) \).
- Vice versa, for every PDA \( A \), there is a CFG \( G \) such that \( L(A) = L(G) \).

The proof is a bit technical and can be found in the appendix. One immediate consequence of Theorem 4.1 is that the intersection of a regular language and a CFL is a CFL.

**Theorem 4.2** If \( K \) is CFL and \( L \) is regular language, then the intersection \( K \cap L \) is CFL.

**Appendix: Proof of Theorem 4.1**

We are going to show that CFG and PDA define precisely the same class of languages. More precisely, we are going to show the following.

- For every CFG \( G \), there is a PDA \( A \) such that \( L(A) = L(G) \).
- For every PDA \( A \), there is a CFG \( G \) such that \( L(A) = L(G) \).

**A From CFG to PDA**

**Allowing the PDA to push a string of symbols.** First, we note that we can modify the definition of CFG to one that allows it to push a string of symbols to its stack. That is, the transitions can be of the form:

\[ (p, x, \text{pop}(y)) \rightarrow (q, \text{push}(z)), \quad \text{where } z \in \Gamma^* \]

Allowing such transition does not change the capability of a CFG. We can add “new” states \( t_1, \ldots, t_m \), where \( m \) is the length of \( z \) and \( z = a_1 \ldots a_m \). Each \( t_i \) is used to push the symbol \( a_i \) into the stack. More formally, the transition \( (p, x, \text{pop}(y)) \rightarrow (q, \text{push}(z)) \) can be replaced with the following transitions:

\[
\begin{align*}
(p, x, \text{pop}(y)) & \rightarrow (t_1, \text{push}(a_1)) \\
(t_1, \varepsilon, \text{pop}(\varepsilon)) & \rightarrow (t_2, \text{push}(a_2)) \\
(t_2, \varepsilon, \text{pop}(\varepsilon)) & \rightarrow (t_3, \text{push}(a_3)) \\
& \vdots \vdots \\
(t_{i-1}, \varepsilon, \text{pop}(\varepsilon)) & \rightarrow (t_i, \text{push}(a_i)) \\
& \vdots \vdots \\
(t_{m-1}, \varepsilon, \text{pop}(\varepsilon)) & \rightarrow (t_m, \text{push}(a_m)) \\
(t_m, \varepsilon, \text{pop}(\varepsilon)) & \rightarrow (q, \text{push}(\varepsilon))
\end{align*}
\]
Left-most substitution property. Let $G = \langle \Sigma, V, R, S \rangle$ be a CFG. Let $z_1 \Rightarrow z_2 \Rightarrow \cdots \Rightarrow z_n$ be a derivation. We say that the derivation has the left-most substitution property, if for every $i \in \{1, \ldots, n-1\}$, $z_i \Rightarrow z_{i+1}$ is obtained by applying a rule $A \rightarrow w$, where $A$ is the left most variable in $z_i$. Intuitively, it means that we only substitute the left-most variable in our derivation.

For example, suppose we have grammars with the following rules: $A \rightarrow aABa, A \rightarrow SS, B \rightarrow aab, B \rightarrow SA$. The derivation $aABAa\Rightarrow aSSBAaa$ has the left-most substitution property, because we substitute the left-most variable which is $A$. On the other hand, $aABAa \Rightarrow aASAAaa$ does not, because we substitute variable $B$, which is not the left-most variable, and $aABAa \Rightarrow aABSSaa$ does not either, because we substitute the second $A$, which is not the left-most.

Remark 4.3 Without loss of generality, we only need to consider derivations that has left-most substitution property, by simply substituting the left-most variable first.

Constructing a PDA from a CFG. Let $G = \langle \Sigma, V, R, S \rangle$ be a CFG. Consider the following PDA $A = \langle \Sigma, \Gamma, Q, q_0, F, \delta \rangle$ where each component is as follows.

- $\Gamma = \Sigma \cup V \cup \{\bot\}$.
- $Q = \{p, q\}$
- $p$ is the initial state.
- $F = \{q\}$.
- $\delta$ consists of the following transitions. (Here $w^r$ denotes the reverse of $w$.)
  - $(p, \varepsilon, \text{pop}(\varepsilon)) \rightarrow (q, \text{push}(S))$.
  - $(q, a, \text{pop}(a)) \rightarrow (q, \text{push}(\varepsilon))$, for every $a \in \Sigma$.
  - $(q, \varepsilon, \text{pop}(A)) \rightarrow (q, \text{push}(s^r))$, for every rule $A \rightarrow s \in R$.

Notice that in the first and third transitions, $A$ pushes a string of symbols to its stack.

We will show that $L(G) = L(A)$. First, we show the following lemma.

Lemma 4.4 For every derivation:

$$u \Rightarrow \tilde{u}$$

there is a run of $A$ on $w$:

$$(q, v) \vdash_w^* (q, \tilde{v}),$$

where

$$v^r = u \quad \text{and} \quad w\tilde{v}^r = \tilde{u} \quad (\dagger)$$

Proof. Suppose $u \Rightarrow^* \tilde{u}$. Let $m$ be the length of the derivation, which is denoted as follows.

$$u_0 \Rightarrow u_1 \Rightarrow \cdots \Rightarrow u_{m-1} \Rightarrow u_m \quad (1)$$

where $u_0 = u$ and $u_m = \tilde{u}$. We will prove the lemma by induction on $m$.

The base case is $m = 0$. In this case the derivation is a trivial derivation:

$$u_0 \Rightarrow^* u_0$$
The run:

\[(q, v) \vdash^* (q, v), \quad \text{where } v^r = u_0\]
satisfies property \((\dagger)\).

For the induction hypothesis, we assume that the lemma holds when the length of the derivation is \(m - 1\). We will prove the derivation of length \(m\) case for the induction step.

Consider a derivation as in \((\dagger)\). We may assume that it has the left-most substitution property. Suppose the derivation \(u_0 \Rightarrow u_1 \Rightarrow \cdots \Rightarrow u_m\) is obtained by applying the rule \(A \rightarrow s\), where variable \(A\) is the left-most variable in \(u_0\). So, we can denote \(u_0\) by \(xAy\), where \(x \in \Sigma^*\) and hence, \(u_1 = xsy\). Since \(x\) contains only terminals, using transitions of the form \((q, a, \text{pop}(a)) \rightarrow (q, \text{push}(\varepsilon))\), where \(a \in \Sigma\), there is a run:

\[(q, y^rAx^r) \vdash^* (q, y^rA)\]  \((2)\)

By our construction of \(A\), there is a transition \((q, \varepsilon, \text{pop}(A)) \rightarrow (q, \text{push}(s^r))\). So, we have:

\[(q, y^rA) \vdash_\varepsilon (q, y^rs^r)\]  \((3)\)

Moreover, \(x\) will not change during the derivation \(u_0 \Rightarrow u_1 \Rightarrow \cdots \Rightarrow u_m\) (because \(x\) contains only terminals). Thus, the string \(u_i\) has prefix \(x\), for every \(i \in \{1, \ldots, m\}\). That is, \(u_i = xu'_i\), for some \(u'_i \in (\Sigma \cup V)^*\). So, the derivation is of the form:

\[xAy \Rightarrow xu'_1 \Rightarrow \cdots \Rightarrow xu'_m.\]

Ignoring the first part of the derivation, we have:

\[xu'_1 \Rightarrow \cdots \Rightarrow xu'_m,\]

which means we also have derivation (since \(x\) contains only terminals):

\[u'_1 \Rightarrow \cdots \Rightarrow u'_m,\]

which has length \(m - 1\). By the induction hypothesis, there is a run:

\[(q, z_1) \vdash^*_t (q, z_2), \quad \text{where } z_1^r = u'_1 \text{ and } tz_2^r = u'_m+1.\]  \((4)\)

Now, recall that \(u_1 = xu'_1 = xsy\), and hence, \(z_1 = y^rs^r\). Combining the runs \((2)\), \((3)\) and \((4)\), we have the following run:

\[(q, y^rAx^r) \vdash^*_x (q, y^rA) \vdash_\varepsilon (q, y^rs^r) \vdash^*_t (q, z_2)\]

which can be abbreviated as:

\[(q, y^rAx^r) \vdash^*_xt (q, z_2).\]

Since \(tz_2^r = u'_m\), we have \(xtz_2^r = xu'_m = u_m\). Since \(u_0 = xAy\), this is the desired property \((\dagger)\).

This completes the proof of Lemma 4.4.

The next lemma is the converse direction of Lemma 4.4.

**Lemma 4.5** For every run of \(A\):

\[(q, v) \vdash^*_w (q, \tilde{v}),\]

there is a derivation:

\[u \Rightarrow^* \tilde{u}\]

such that:

\[v^r = u \quad \text{and} \quad w\tilde{v}^r = \tilde{u}\]  \((*)\)

\[11\]
**Proof.** Suppose we have a run \((q, v) \vdash^*_w (q, \tilde{v})\). Let \(n\) be the length of the run, which is denoted as follows.

\[
(q, v_0) \vdash b_1 (q, v_1) \vdash b_2 \cdots \vdash b_n (q, v_n),
\]

where \(v_0 = v, v_n = \tilde{v}\) and \(b_1 \cdots b_n = w\). We will prove the lemma by induction on \(n\).

The base case is \(n = 0\). In this case, the run is a trivial run:

\[
(q, v_0) \vdash^* (q, v_0)
\]

Thus, the trivial derivation:

\[ u \Rightarrow^* u \quad \text{where } u = v_0^r \]

satisfies the desired property \((\star)\).

For the induction hypothesis, we assume that the lemma holds when the length of the run is \(n - 1\). We will prove the case when the length of the run is \(n\) for the induction step.

We consider a run of length \(n\) as in (5). We consider the step \((q, v_0) \vdash b_1 (q, v_1)\). There are two cases:

- **\(b_1 \neq \varepsilon\), i.e., \(b_1\) is a symbol from \(\Sigma\).**

  By the construction of the PDA \(A\), it has to use the transition \((q, b_1, \text{pop}(b_1)) \rightarrow (q, \text{push}(\varepsilon))\), which means that the top of the stack \(v_0\) is \(b_1\). Therefore, \(v_0 = v_1 b_1\).

  Now consider the run:

  \[
  (q, v_1) \vdash b_2 \cdots \vdash b_n (q, v_n).
  \]

  This run is of length \(n - 1\). By the induction hypothesis, we have a derivation:

  \[ u_1 \Rightarrow^* u_2 \quad \text{where } u_1 = v_1^r \text{ and } u_2 = b_2 \cdots b_n v_n^r. \]

  Adding \(b_1\) in front of \(u_1\) and \(u_2\), we have:

  \[ b_1 u_1 \Rightarrow^* b_1 u_2 \]

  Now, \(b_1 u_1 = b_1 v_1^r = v_0^r\) and \(b_1 u_2 = b_1 b_2 \cdots b_n v_n^r\), which is the desired property \((\star)\).

- **\(b_1 = \varepsilon\).**

  By the construction of the PDA \(A\), it has to use the transition \((q, \varepsilon, \text{pop}(A)) \rightarrow (q, \text{push}(s^r))\), for some rule \(A \rightarrow s\). This means that the top of the stack \(v_0\) is \(A\). We denote by \(v_0 = yA\) and \(v_1 = ys^r\), for some \(y\).

  Consider the run:

  \[
  (q, v_1) \vdash b_2 \cdots \vdash b_n (q, v_n).
  \]

  This run has length \(n - 1\). By the induction hypothesis, there is a derivation:

  \[ u_1 \Rightarrow^* u_2 \quad \text{where } u_1 = v_1^r \text{ and } u_2 = b_2 \cdots b_n v_n^r. \]

  Since \(v_1 = ys^r\), and hence, \(u_1 = sy^r\), we have a derivation:

  \[ Ay^r \Rightarrow u_1 \Rightarrow^* u_2. \]

  By our notation, \(v_0 = yA\) and \(u_2 = b_2 \cdots b_n v_n^r\), which is the desired property \((\star)\).
This completes our proof of Lemma 4.5.

Using the two lemmas above, we can show that \( L(A) = L(G) \) as stated below.

**Theorem 4.6** \( L(G) = L(A) \).

**Proof.** We first prove \( L(G) \subseteq L(A) \). Let \( S \Rightarrow^* w \), where \( w \in \Sigma^* \). By Lemma 4.4 there is a run:

\[
(q, S) \Rightarrow^* (q, y), \quad \text{where } xy^r = w.
\]

Since \( w \) contains only symbols from \( \Sigma \), so does \( y \). If \( y \neq \varepsilon \), using the transitions of the form \((q, a, \text{pop}(a)) \rightarrow (q, \text{push}(\varepsilon))\), we can extend the run until \( y = \varepsilon \), which means that \( x = w \).

By the construction of \( A \), we have the following accepting run on \( w \):

\[
(p, \varepsilon) \Rightarrow^* (q, S) \Rightarrow^* (q, \varepsilon).
\]

Thus, \( w \in L(A) \).

Now we prove \( L(A) \supseteq L(G) \). Let \( w \in L(A) \). So, there is an accepting run of \( A \) on \( w \), which must start from the initial state \( p \) and end with the accepting state \( q \) after finishes reading the input word \( w \). Thus, it has to start by using the transition \((p, \varepsilon, \text{pop}(\varepsilon)) \rightarrow (q, \text{push}(S))\). This means the run is of the form:

\[
(p, \varepsilon) \Rightarrow^* (q, S) \Rightarrow^* (q, \varepsilon).
\]

By Lemma 4.5 there is a derivation:

\[
u_1 \Rightarrow^* u_2, \quad \text{where } u_1 = S \text{ and } u_2 = w.\]

Thus, \( S \Rightarrow^* w \) and \( w \in L(G) \). This completes the proof of Theorem 4.6.

**B From PDA to CFG**

Let \( A = (\Sigma, \Gamma, Q, q_0, F, \delta) \) be a PDA. Without loss of generality, we can assume the following.

- It has only one final state, say \( q_f \). That is, \( F = \{q_f\} \).
- The stack is empty before accepting an input word. More precisely, on every word \( w \), if \( A \) accepts \( w \), there is an accepting run of \( A \) on \( w \) from the initial configuration \((q_0, \varepsilon)\) to a final configuration \((q_f, \varepsilon)\) where the content of the stack is empty.
- In each transition, \( A \) either pushes a symbol into the stack or pops one from the stack, but it cannot do both. More precisely, every transition can only be of the forms:

\[
(p, x, \text{pop}(y)) \rightarrow (q, \text{push}(\varepsilon))
\]

\[
(p, x, \text{pop}(\varepsilon)) \rightarrow (q, \text{push}(\varepsilon))
\]

Consider the following CFG \( G = (\Sigma, V, R, S) \) where each component is as follows.

- \( V = \{A_{p,q} \mid p, q \in Q\} \).
- \( A_{q_0,q_f} \) is the start variable.
- \( R \) consists of the following rules:
- For every state \( p, q, r, s \in Q \) and every symbol \( z \in \Gamma \) and every symbol \( a, b \in \Sigma \cup \{ \varepsilon \} \), if the following transitions are in \( \delta \):

\[
(p, a, \text{pop}(\varepsilon)) \rightarrow (r, \text{push}(z))
\]

\[
(s, b, \text{pop}(z)) \rightarrow (q, \text{push}(\varepsilon))
\]

then the following rule is in \( R \):

\[
A_{p,q} \rightarrow a A_{r,s} b \quad (\mathcal{R}1)
\]

- For every state \( p, q, r \in Q \) and every symbol \( a \in \Sigma \cup \{ \varepsilon \} \), if the following transition is in \( \delta \):

\[
(p, a, \text{pop}(\varepsilon)) \rightarrow (r, \text{push}(\varepsilon))
\]

then the following rule is in \( R \):

\[
A_{p,q} \rightarrow a A_{r,q} \quad (\mathcal{R}2)
\]

- For every state \( p, q, r \in Q \), we have the following rule in \( R \):

\[
A_{p,q} \rightarrow A_{p,r} A_{r,q} \quad (\mathcal{R}3)
\]

- For every \( p \in Q \), we have the following rule in \( R \):

\[
A_{p,p} \rightarrow \varepsilon \quad (\mathcal{R}4)
\]

We will show that \( L(A) = L(G) \). First, we prove the following lemma.

**Lemma 4.7** For every derivation:

\[
A_{p,q} \Rightarrow^* w, \quad \text{where } w \in \Sigma^*
\]

there is a run:

\[
(p, \varepsilon) \vdash^*_w (q, \varepsilon)
\]

**Proof.** Suppose \( A_{p,q} \Rightarrow^* w \), where \( w \in \Sigma^* \). Let the derivation of length \( m \) and denoted as follows.

\[
A_{p,q} \Rightarrow w_1 \Rightarrow \cdots \Rightarrow w_m, \quad \text{where } w_m = w. \quad (6)
\]

We will prove the lemma by induction on \( m \). The base case is \( m = 1 \). So, we have a derivation:

\[
A_{p,q} \Rightarrow w
\]

The only rule that can be used in this case is rule \( \mathcal{R}4 \), which means \( w = \varepsilon \) and \( p = q \). Thus, there is a (trivial) run:

\[
(p, \varepsilon) \vdash^*_\varepsilon (p, \varepsilon)
\]

For the induction hypothesis, we assume the lemma holds for every derivation of length \( \leq m - 1 \). We will prove the case of derivations of length \( m \) for the induction step.

Let the derivation be as in (6). We consider the rules applied on the step \( A_{p,q} \Rightarrow w_1 \). There are three cases:
• The rule applied is \((R_1)\) type of rule, say:

\[ A_{p,q} \rightarrow aA_{r,s}b \]

This means that \(w_1 = aA_{r,s}b\), and hence, every \(w_i\) starts with \(a\) and ends with \(b\), for every \(i \in \{1, \ldots, m\}\). We denote each \(w_i = aw'_ib\) for some \(w'_i\). Thus, there is a derivation:

\[ A_{r,s} \Rightarrow w'_2 \Rightarrow \cdots \Rightarrow w'_m \]

This derivation has length \(m - 1\). By the induction hypothesis, there is a run:

\[ (r, \varepsilon) \vdash w'_{m+1} (s, \varepsilon) \quad (7) \]

Since \(A_{p,q} \rightarrow aA_{r,s}b\) is a rule in \(R\), there are the following two transitions in \(\delta\):

\[ (p, a, \text{pop}(\varepsilon)) \rightarrow (r, \text{push}(z)) \quad \text{and} \quad (s, b, \text{pop}(z)) \rightarrow (q, \text{push}(\varepsilon)) \quad (8) \]

Now, from the run \((7)\), since the content of the stack never goes "below empty", we also have a run:

\[ (r, z) \vdash w'_{m+1} (s, z) \quad (9) \]

Using the transitions \((8)\), we have:

\[ (p, \varepsilon) \vdash_a (r, z) \quad \text{and} \quad (s, z) \vdash_b (q, \varepsilon) \quad (10) \]

Combining the runs \((9)\) and \((10)\), we have:

\[ (p, \varepsilon) \vdash_a (r, z) \vdash_{w'_{m+1}}^* (s, z) \vdash_b (q, \varepsilon) \]

That is,

\[ (p, \varepsilon) \vdash_{aw'_{m+1}b}^* (q, \varepsilon) \]

Since \(w = aw'_{m+1}b\), the run is as desired.

• The rule applied is \((R_3)\) type of rule, say:

\[ A_{p,q} \rightarrow A_{p,r}A_{r,q}, \quad \text{for some } r \in Q. \]

Thus, \(w_1 = A_{p,r}A_{r,q}\). This means \(w = xy\) where \(A_{p,r} \Rightarrow^* x\) and \(A_{r,q} \Rightarrow^* y\). Both are derivations of length \(\leq m - 1\). By the induction hypothesis, there are the following two runs:

\[ (p, \varepsilon) \vdash_{x}^* (r, \varepsilon) \quad \text{and} \quad (r, \varepsilon) \vdash_{y}^* (q, \varepsilon) \]

Combining these two runs, we have:

\[ (p, \varepsilon) \vdash_{w}^* (r, \varepsilon) \]

• The rule applied is \((R_2)\) type of rule, say:

\[ A_{p,q} \rightarrow aA_{r,q}, \text{ for some } r \in Q \text{ and } a \in \Sigma \]

\[ (p, \varepsilon) \vdash_{aw}^* (r, \varepsilon) \]
Thus, \( w_1 = aA_{r,q} \). This means that every \( w_i \) starts with \( a \), for every \( i \in \{1, \ldots, m\} \). We denote each \( w_i = aw'_i \), for some \( w'_i \). Thus, there is a run:

\[
A_{r,q} \Rightarrow w'_2 \Rightarrow \cdots \Rightarrow w'_m.
\]

The length of this run is \( m - 1 \). By the induction hypothesis, there is a run:

\[
(r, \varepsilon) \vdash_w^* (q, \varepsilon)\tag{11}
\]

Now, since there is a rule \( A_{p,q} \rightarrow aA_{r,q} \), there is the following transition in \( \delta \):

\[
(p, a, \text{pop}(\varepsilon)) \rightarrow (r, \text{push}(\varepsilon))
\]

Using this transition, we have:

\[
(p, \varepsilon) \vdash_a (r, \varepsilon)
\]

Combining this with run (11), we have run:

\[
(p, \varepsilon) \vdash_a (r, \varepsilon) \vdash_w^* (q, \varepsilon)
\]

Thus, there is a run:

\[
(p, \varepsilon) \vdash_{aw'_m}^* (q, \varepsilon)
\]

This run is as required.

This completes the proof of Lemma 4.7.

The following lemma is the converse direction of the previous lemma.

**Lemma 4.8** For every run:

\[
(p, \varepsilon) \vdash_w^* (q, \varepsilon)
\]

there is a derivation:

\[
A_{p,q} \Rightarrow^* w
\]

**Proof.** Suppose there is a run \( (p, \varepsilon) \vdash_w^* (q, \varepsilon) \). Let the run has length \( n \), denoted as follows.

\[
(p_0, v_0) \vdash_{b_1} (p_1, v_1) \vdash_{b_2} \cdots \vdash_{b_n} (p_n, v_n)\tag{12}
\]

where \( p_0 = p, v_0 = \varepsilon, p_n = q, v_n = \varepsilon \) and \( b_1 \cdots b_n = w \). We will prove the lemma by induction on \( n \).

The base case is \( n = 0 \). In this case the run is a trivial run:

\[
(p_0, \varepsilon) \vdash_{\varepsilon}^* (p_0, \varepsilon)
\]

Applying the \( \mathcal{R}_4 \) type of rule \( A_{p_0,p_0} \rightarrow \varepsilon \), we have a derivation:

\[
A_{p_0,p_0} \Rightarrow \varepsilon
\]

For the induction hypothesis, we assume that the lemma holds for run of length \( \leq n - 1 \). We will now prove it for runs of length \( n \) for the induction step.

Consider a run of length \( n \) as in (12). We consider the transition used in the step \( (p_0, v_0) \vdash_{b_1} (p_1, v_1) \). There are two cases:
• The transition used is \((p_0, b_1, \text{pop}(\varepsilon)) \rightarrow (p_1, \text{push}(\varepsilon))\), i.e., the PDA \(A\) pops and pushes nothing to its stack. By definition of the CFG \(G\), there is a \([R2]\) type of rule:

\[
A_{p_0,p_n} \rightarrow b_1 A_{p_1,p_n}
\]  
(13)

Applying the induction hypothesis on the run:

\[(p_1, v_1) \vdash b_2 \cdots \vdash b_n (p_n, v_n)\]

there is a derivation:

\[
A_{p_1,p_n} \Rightarrow^* b_2 \cdots b_n
\]

Moreover, using rule \([R3]\), we have derivation:

\[
A_{p_0,p_n} \Rightarrow b_1 A_{p_1,p_n} \Rightarrow^* b_1 b_2 \cdots b_n
\]

as desired.

• The transition used is of the form:

\[
(p_0, b_1, \text{pop}(\varepsilon)) \rightarrow (p_1, \text{push}(z))\]

for some \(z \in \Gamma\).

There are two more cases here.

– For some \(j \in \{2, \ldots, n\}\), \(v_j = \varepsilon\).

Thus, there is a run:

\[(p_0, v_0) \vdash^*_x (p_j, v_j) \vdash^*_y (p_n, v_n)\]

where \(w = xy\).

By induction hypothesis, there are derivations:

\[
A_{p_0,p_j} \Rightarrow^* x \quad \text{and} \quad A_{p_j,p_n} \Rightarrow^* y
\]

By construction of \(G\), there is a \([R3]\) type of rule:

\[
A_{p_0,p_n} \rightarrow A_{p_0,p_j} A_{p_j,p_n}.
\]

Using this rule, we have the following derivation:

\[
A_{p_0,p_n} \Rightarrow A_{p_0,p_j} A_{p_j,p_n} \Rightarrow^* xy
\]

This is the desired derivation, since \(w = xy\).

– For every \(i \in \{1, \ldots, n - 1\}\), \(v_i \neq \varepsilon\).

This means that \(z\) is popped from \(v_{n-1}\) to obtain \(v_n\) and there are the following transitions:

\[(p_0, a, \text{pop}(\varepsilon)) \rightarrow (p_1, \text{push}(z)) \quad \text{and} \quad (p_{n-1}, b, \text{pop}(\varepsilon)) \rightarrow (p_n, \text{push}(\varepsilon))\]  
(14)

where \(b_1 = a\) and \(b_n = b\). Since \(v_n = \varepsilon\), this means \(v_{n-1} = z\). In particular, every \(v_i\) starts with \(z\), for every \(i \in \{1, \ldots, n - 1\}\). We denote by \(v_i = zw'_i\), for some \(i \in \{1, \ldots, n - 1\}\). This means there is a run:

\[(p_1, z) \vdash^*_w (p_{n-1}, z)\]

where \(w = aw'b\)

Since during this run, \(z\) is untouched, we have a run:

\[(p_1, \varepsilon) \vdash^*_w (p_{n-1}, \varepsilon)\]
This run has length $\leq n - 1$. By the induction hypothesis, there is a derivation:

$$A_{p1,p_{n-1}} \Rightarrow^* w'$$

Due to the transitions in (14), there is a ($R_1$) type of rule:

$$A_{p0,p_n} \rightarrow aA_{p1,p_{n-1}}b$$

Using this rule, there is a derivation:

$$A_{p0,p_n} \Rightarrow aA_{p1,p_{n-1}}b \Rightarrow^* aw'b.$$

Since $aw'b = w$, this derivation is as desired.

This completes the proof of Lemma 4.8.

**Theorem 4.9** $L(A) = L(G)$.

**Proof.** We first prove $L(G) \supseteq L(A)$. Let $w \in L(A)$. So, there is an accepting run of $A$ on $w$, which must start from the initial state $q_0$ and end with the accepting state $q_f$:

$$(q_0, \varepsilon) \vdash^*_w (q_f, \varepsilon)$$

By Lemma 4.8 we have:

$$A_{q_0,q_f} \Rightarrow^* w.$$  

Thus, $w \in L(G)$.

Now we prove that $L(G) \subseteq L(A)$. Let $w \in L(G)$. Thus,

$$A_{q_0,q_f} \Rightarrow^* w.$$  

By Lemma 4.7 there is a run:

$$(q_0, \varepsilon) \vdash^*_w (q_f, \varepsilon).$$

Since this is an accepting run, $w \in L(A)$. This completes the proof of Theorem 4.9.