Pushdown Automata - I (part I)

Prof. Dan A. Simovici

UMB

1. Nondeterministic Pushdown Automata

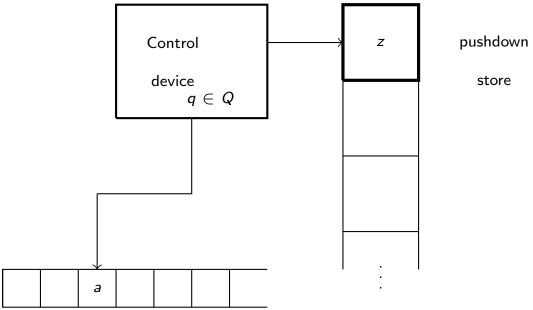
- Pushdown automata provide abstract models of computation. They play the same role for the class of context-free languages that finite automata play for regular languages: the class of languages accepted by pushdown automata is precisely the class $\mathcal{L}_2$ of context-free languages.
- Unlike finite automata, deterministic pushdown automata are weaker than non-deterministic ones; i.e., they accept only languages belonging to a strict subclass of $\mathcal{L}_2$.

- Just like a finite automaton, a *pushdown automaton* (pda) has a finite set of states and a one-way, read-only input tape, divided into cells each of which contains a symbol of an alphabet A, referred to as the input alphabet.
- The pushdown model adds a pushdown store, that can be conceptualized as a tape that has a beginning but is infinitely long. It, too, is divided into cells, and each cell that is in use contains a symbol from an alphabet Z, referred to as the pushdown alphabet. The pushdown store is thought of as being oriented vertically, with the used portion at the top.
- The automaton can read and write only the top-most cell, referred to as the top of the pushdown store.

As in the case of nondeterministic finite automata, a pda accepts a word if there is a sequence of choices that allows the suitably initialized pda to consume the word and end in a state that is an element of a specified set of accepting states.

The set of words accepted by a pda is referred to as the language accepted by that pda. We shall show an exact correspondence between the class of languages accepted by pdas and the class of context-free languages.

Components of a Pushdown Automaton



Let $\mathcal{P}_f(M)$ be the set of all finite subsets of a set M.

Definition

A *pushdown automaton* (pda) is a 7-tuple

$$\mathcal{M} = (A, Z, Q, \delta, q_0, z_0, F),$$

where A is the *input alphabet*, Z is the *alphabet of the pushdown store*,

$$\delta : Z \times Q \times (A \cup \{\lambda\}) \longrightarrow \mathcal{P}_f(Z^* \times Q)$$

is the *transition function*, q_0 is the *initial state*, z_0 is the *start symbol*, and F is the *set of final states*. We assume that $Q \cap A = Q \cap Z = \emptyset$.
We refer to the pairs $(q', w) \in \delta(z, q, a)$ as the *transitions of the triple* (z, q, a).
The transitions of a triple (z, q, λ) (where nothing is read from the input tape) are referred to as (z, q)-*null transitions*.

The Working of a PDA

When scanning the input symbol a in the state q and reading the symbol z from the top of the pushdown store, the automaton selects a pair $(w, q') \in Z^* \times Q$ from the set $\delta(z, q, a)$; after that:

- the state changes to q' and the top symbol of the pushdown store is replaced by the word w;
- If $w = \lambda$, this amounts to popping the top symbol out of the top cell; if $|w| > 1$, the replacement of z by w causes the symbols located below the top cell to be pushed downwards by $|w| - 1$ cells in order to accommodate the symbols of the word w.

Example

If the state of the pda is q, the content of the pushdown store is $w'z$, where z is the symbol at the top, and the pda chooses the pair $(z_{j_0} \cdots z_{j_{m-1}}, q')$ from $\delta(z, q, a)$, then:

- the new content of the pushdown store will be $w'z_{j_0} \cdots z_{j_{m-1}}$;
- and the new state will be q';
- the symbol at the top of the stack will then be $z_{j_{m-1}}$.

A pda is a nondeterministic device, because at any given moment a pda may chose one among several transitions (z, q') for its next step.

Example

Consider the pushdown automaton

$$\mathcal{M} = (\{a, b\}, \{z_0, z_1, z_2, z_3\}, \{q_0, q_1, q_2\}, \delta, q_0, z_0, \{q_2\}),$$

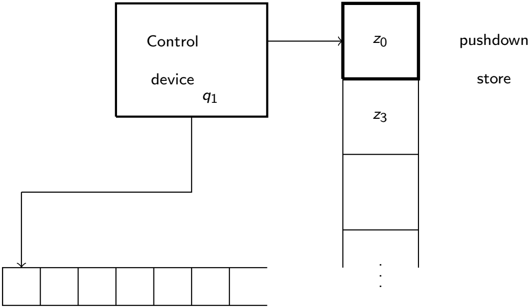
where δ is defined by the table

Top z	State q	Input a	Transition Function $\delta(z, q, a)$
z_0	q_0	λ	$\{(z_3 z_0), q_1\}$
z_0	q_1	λ	$\{(\lambda, q_1), (z_1 z_0 z_1, q_1), (z_2 z_0 z_2, q_1)\}$
z_1	q_1	a	$\{(\lambda, q_1)\}$
z_2	q_1	b	$\{(\lambda, q_1)\}$
z_3	q_1	λ	$\{(z_3, q_2)\}$

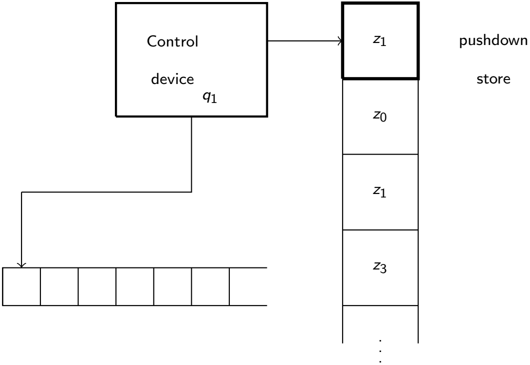
For triples (z, q, a) that do not appear in this table we assume that $\delta(z, q, a) = \emptyset$.

(Example cont'd)

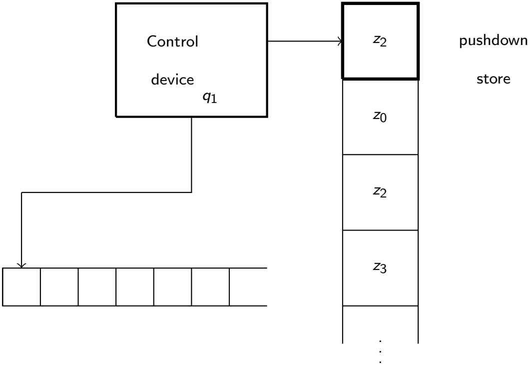
$\mathcal{M}$ begins by writing $z_3 z_0$ onto the stack. Thus, z_3 is placed at the bottom of the stack. After reading through the entire input, the pda will have to find z_3 at the top of the stack in order to go into state q_2, which is the only accepting state.



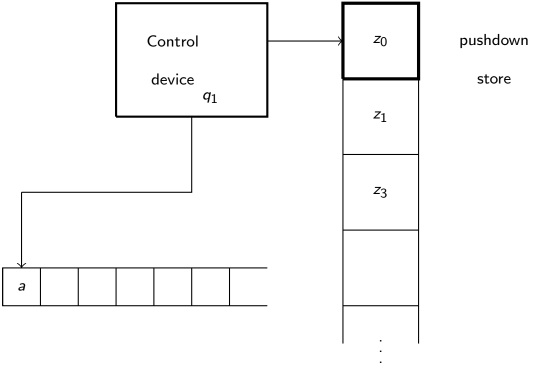
When z_0 is at the top of the stack, the pda guesses whether the next symbol is a or b. If it guesses a, it pushes $z_1 z_0 z_1$ onto the stack.



If it guesses b, it pushes $z_2 z_0 z_2$ onto the stack.



It then verifies its guess by examining the next symbol. If it made the "right guess", then it has left a trace of what that symbol was on the stack. For example, if it guessed that the next symbol is a and z_1 is at the top of the stack, the next configuration will pop z_1:



If it made the "wrong guess", it cannot proceed.

(Example cont'd)

- Over time, this records the symbols read from the input with the most recent nearest the top. In other words, reading the stack from top to bottom would allow one to reconstruct the input word in reverse order.
- This process continues until the pda guesses that it has reached the mid-point of the input word. At that time, it simply pops the z_0 off the stack.
- Now it reads through the remaining portion of the input, checking that what is there is the reverse of what was encountered in the first half, as only that input will match the trace left in the stack; i.e., if the word x was read from the input tape before the pda guessed it had reached the middle, only x^R will allow it to pop each symbol off the stack as the pda processes the remaining input word. When it exhausts the input, z_3 should be at the top of the stack, allowing the pda to accept the word.

- There are many places where the pda must make guesses, but if it is possible to make a sequence of guesses that allows the pda to read through the word on its input tape and end up in an accepting state, then the pda accepts the word.
- Only if no possible sequence of guesses permits the pda to accept the word does it reject the word. In this case, the language accepted by the pda is $\{xx^R \mid x \in \{a, b\}^*\}$.

Definition

Let $\mathcal{M} = (A, Z, Q, \delta, q_0, z_0, F)$ be a pda. The set $\mathcal{I}(\mathcal{M})$ of *instantaneous descriptions of* $\mathcal{M}$ is the set $Z^* \times Q \times A^*$.
A pda $\mathcal{M}$ *is described by* $(w, q, u) \in \mathcal{I}(\mathcal{M})$ if $\mathcal{M}$ is in the state q, u is the portion of input that is still to be read, and w is the content of the pushdown store, with the rightmost symbol of w at the top of the pushdown store.
Any instantaneous description of the form (z_0, q_0, u) is an *initial instantaneous description of* $\mathcal{M}$. □

Definition

Let $\mathcal{M} = (A, Z, Q, \delta, q_0, z_0, F)$ be a pda. The binary relation $\underset{\mathcal{M}}{\vdash}$ on the set $\mathcal{I}(\mathcal{M})$ is given by $(w'z, q, au) \underset{\mathcal{M}}{\vdash} (w'w, p, u)$ if $(w, p) \in \delta(z, q, a)$, for $z \in Z$, $q \in Q$, and $a \in A \cup \{\lambda\}$.
When $\mathcal{M}$ is obvious from context we may simply write $\vdash$ for $\underset{\mathcal{M}}{\vdash}$.

The n^{th} power of $\underset{\mathcal{M}}{\vdash}$ is denoted, as usual, by $\overset{n}{\underset{\mathcal{M}}{\vdash}}$ for $n \in \mathbb{N}$; also, the transitive closure of $\underset{\mathcal{M}}{\vdash}$ is denoted by $\overset{+}{\underset{\mathcal{M}}{\vdash}}$ and the reflexive and transitive closure of the same relation is denoted by $\overset{*}{\underset{\mathcal{M}}{\vdash}}$.

A pda may have instantaneous descriptions (w, q, u) such that there are no instantaneous descriptions (w_1, q_1, u_1) such that $(w, q, u) \underset{\mathcal{M}}{\vdash} (w_1, q_1, u_1)$.

We refer to (w, q, u) as a *blocking instantaneous description*.
For instance, such instantaneous descriptions occur when the pushdown store is empty.

Definition

A *computation* in a pda $\mathcal{M}$ is a sequence $(c_0, \ldots, c_n)$ of instantaneous descriptions of $\mathcal{M}$ such that $c_i \underset{\mathcal{M}}{\vdash} c_{i+1}$ for $0 \leq i \leq n-1$.

We will denote the above computation by $c_0 \underset{\mathcal{M}}{\vdash} \cdots \underset{\mathcal{M}}{\vdash} c_n$.

Example

Let $\mathcal{M}$ be the pda introduced previously. We have

$$\begin{aligned}
&(z_0, q_0, abbbba) \vdash (z_3 z_0, q_1, abbbba) \vdash (z_3 z_1 z_0 z_1, q_1, abbbba) \\
&\vdash (z_3 z_1 z_0, q_1, bbbba) \vdash (z_3 z_1 z_2 z_0 z_2, q_1, bbbba) \vdash (z_3 z_1 z_2 z_0, q_1, bbba) \\
&\vdash (z_3 z_1 z_2 z_2 z_0 z_2, q_1, bbba) \vdash (z_3 z_1 z_2 z_2 z_0, q_1, bba) \\
&\vdash (z_3 z_1 z_2 z_2, q_1, bba) \vdash (z_3 z_1 z_2, q_1, ba) \\
&\vdash (z_3 z_1, q_1, a) \vdash (z_3, q_1, \lambda) \vdash (z_3, q_2, \lambda).
\end{aligned}$$

Therefore, we can write $(z_0, q_0, abbbba) \overset{*}{\underset{\mathcal{M}}{\vdash}} (z_3, q_2, \lambda)$.

(Example cont'd)

- When $\mathcal{M}$ reaches the state q_1 and z_0 is at the top of the pushdown store, there are three possible moves, namely (λ, q_1), $(z_1 z_0 z_1, q_1)$, or $(z_2 z_0 z_2, q_1)$. None of these involves reading an input symbol.
- If $\mathcal{M}$ makes the "wrong guess" when the instantaneous description $(z_3 z_1 z_2 z_0, q_1, bbba)$ is reached and makes use of the pair $(z_1 z_0 z_1, q_1)$ instead of $(z_2 z_0 z_2, q_1)$, the next instantaneous description will be $(z_3 z_1 z_2 z_1 z_0 z_1, q_1, bbba)$.
- A symbol z_1 cannot be popped out of the pushdown store unless $\mathcal{M}$ reads a from the input. Since only one symbol a remains in the input, at most one z_1 can be popped off, and thus it would be impossible to get z_3 at the top of the pushdown store. This would prevent $\mathcal{M}$ from reaching the final state q_2.

Theorem

Let $\mathcal{M} = (A, Z, Q, \delta, q_0, z_0, F)$ be a pda.

1. *If $(z_{i_0} \cdots z_{i_{n-1}}, p, t) \overset{*}{\vdash} (\lambda, p', \lambda)$, then we can write t as a product of n words $t = t_{n-1} \cdots t_0$ such that*

$$\begin{array}{rcl} (z_{i_{n-1}}, p_0, t_{n-1}) & \overset{*}{\vdash} & (\lambda, p_1, \lambda) \\ (z_{i_{n-2}}, p_1, t_{n-2}) & \overset{*}{\vdash} & (\lambda, p_2, \lambda) \\ & \cdots & \\ (z_{i_0}, p_{n-1}, t_0) & \overset{*}{\vdash} & (\lambda, p_n, \lambda), \end{array}$$

   where $p = p_0, p_1, \ldots, p_n = p'$ is a sequence of states in Q.

2. $(w, q, x) \overset{*}{\vdash} (w', q', \lambda)$ *for some $q, q' \in Q$, $w, w' \in Z^*$, and $x \in A^*$ if and only if $(w, q, xy) \overset{*}{\vdash} (w', q', y)$ for every $y \in A^*$.*

3. *If $(w, q, xy) \overset{*}{\vdash} (w', q', y)$, then $(w_1 w, q, xy) \overset{*}{\vdash} (w_1 w', q', y)$ for each $w_1 \in Z^*$.*

Proof of first part

The argument for the first part of the theorem is by induction on the length n of the initial content of the pushdown store $w = z_{i_0} \cdots z_{i_{n-1}}$, where $n \geq 1$. The base case, $n = 1$ is immediate.
Suppose that the statement holds for $|w| \leq n-1$, and let $w = z_{i_0} \cdots z_{i_{n-1}}$ be a word of length n. The symbol $z_{i_{n-1}}$ at the top of the pushdown store must eventually be erased. If t_{n-1} is the prefix of input word t that causes the erasure of $z_{i_{n-1}}$, then we can write $t = t_{n-1}t'$, where

$$\begin{aligned} (z_{i_{n-1}}, p_0, t_{n-1}) & \overset{*}{\vDash} (\lambda, p_1, \lambda) \\ (z_{i_0} \cdots z_{i_{n-2}}, p_1, t') & \overset{*}{\vDash} (\lambda, p', \lambda) \end{aligned}$$

for some state $p_1 \in Q$.

(Proof cont'd)

The inductive hypothesis implies that t' can be written as a product of $n-1$ words $t' = t_{n-2} \dots t_0$ such that

$$\begin{array}{rcl} (z_{i_{n-2}}, p_1, t_{n-2}) & \overset{*}{\vDash} & (\lambda, p_2, \lambda) \\ & \dots & \\ (z_{i_0}, p_{n-1}, t_0) & \overset{*}{\vDash} & (\lambda, p_n, \lambda), \end{array}$$

where $p_1, p_2, \dots, p_{n-1} = p'$ is a sequence of states of $\mathcal{M}$. Combining this fact with the previous observation gives the desired conclusion.

As in the case of nondeterministic finite automata, pdas accept languages by reading through their input words, making guesses as necessary along the way, and ending in a final, accepting state. This is formalized in the next definition.

Definition

The *language accepted by the pda* $\mathcal{M} = (A, Z, Q, \delta, q_0, z_0, F)$ is the language

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

The pda $\mathcal{M}, \mathcal{M}'$ are said to be *equivalent* if $L(\mathcal{M}) = L(\mathcal{M}')$.

Example

The language accepted by $\mathcal{M}$, the pda previously introduced, is $L(\mathcal{M}) = \{xx^R \mid x \in \{a,b\}^*\}$.

We begin by showing that for every $x \in \{a,b\}^*$ we have $xx^R \in L(\mathcal{M})$. To this end, we prove that $(z_0, q_1, xx^R) \overset{*}{\underset{\mathcal{M}}{\vdash}} (\lambda, q_1, \lambda)$. The argument is by induction on the length of x.

For $|x| = 0$, we have $xx^R = \lambda$ and $(z_0, q_1, \lambda) \overset{*}{\underset{\mathcal{M}}{\vdash}} (\lambda, q_1, \lambda)$ in view of the fact that $(q_1, \lambda) \in \delta(z_0, q_1, \lambda)$.

(Example cont'd)

Suppose that our claim holds for words of length n, and let $y \in A^*$ be a word of length $n+1$. We can write either $y = au$, or $y = bv$ for some $u, v \in A^*$ with $|u| = |v| = n$, depending on the first symbol of y. In the first case we can write:

$$
\begin{array}{lll}
(z_0, q_1, yy^R) = (z_0, q_1, auu^R a) & & \\
\quad \vdash & (z_1 z_0 z_1, q_1, auu^R a) & (\text{since } (z_1 z_0 z_1, q_1) \in \delta(z_0, q_1, \lambda)) \\
\quad \vdash & (z_1 z_0, q_1, uu^R a) & (\text{since } (\lambda, q_1) \in \delta(z_1, q_1, a)) \\
\quad \overset{*}{\vdash} & (z_1, q_1, a) & (\text{by the inductive hypothesis}) \\
\quad \vdash & (\lambda, q_1, \lambda) & (\text{since } (\lambda, q_1) \in \delta(z_1, q_1, a)).
\end{array}
$$

The case $y = bv$ can be treated similarly.

(Example cont'd)

Since $(z_3z_0, q_1) \in \delta(z_0, q_0, \lambda)$ we have

$$(z_0, q_0, xx^R) \vdash (z_3z_0, q_1, xx^R),$$

for every $x \in \{a, b\}^*$.
Therefore, by the previous claim, we obtain the computation:

$$\begin{array}{rcl} (z_0, q_0, xx^R) & \vdash & (z_3z_0, q_1, xx^R) \\ & \overset{*}{\vdash} & (z_3, q_1, \lambda) \\ & \vdash & (z_3, q_2, \lambda), \end{array}$$

so $xx^R \in L(\mathcal{M})$ because q_2 is a final state of $\mathcal{M}$.

(Example cont'd)

Conversely, suppose that $t \in L(\mathcal{M})$. We have

$$(z_0, q_0, t) \overset{*}{\vdash} (w, q_2, \lambda),$$

for some $w \in Z^*$. The definition of $\mathcal{M}$ implies that this computation must be written as

$$(z_0, q_0, t) \vdash (z_3 z_0, q_1, t) \overset{*}{\vdash} (w, q_2, \lambda),$$

since this is the single means of switching from the state q_0 into the state q_1, which, in turn, is the unique state of $\mathcal{M}$ that precedes the final state q_2.

To reach the final state q_2, the pda must have the symbol z_3 at the top of the pushdown store. Thus, the instantaneous description that precedes (w, q_2, λ) in the previous computation necessarily has the form (z_3, q_1, λ). This implies that $w = z_3$, so we have

$$(z_0, q_0, t) \vdash (z_3 z_0, q_1, t) \overset{*}{\vdash} (z_3, q_1, \lambda) \vdash (z_3, q_2, \lambda).$$

In turn, this means that we have the computation

$$(z_0, q_1, t) \overset{*}{\vdash} (\lambda, q_1, \lambda),$$

during which $\mathcal{M}$ remains in the state q_1. We prove by induction on $k = |t|$ that $t = xx^R$ for some $x \in \{a, b\}^*$.

(Example cont'd)

If $k = 0$, then $t = \lambda$, so $t = \lambda\lambda^R$. In this case, $\mathcal{M}$ chooses the pair $(\lambda, q_1) \in \delta(z_0, q_1, \lambda)$, and we have

$$(z_0, q_1, t) \vdash (\lambda, q_1, t);$$

thus, the statement holds.

Suppose that the statement holds for words t of length less than k, and let $t \in A^*$ be a word of length k. If $\mathcal{M}$ chooses the pair $(z_1 z_0 z_1, q_1) \in \delta(z_0, q_1, \lambda)$, then

$$(z_0, q_1, t) \vdash (z_1 z_0 z_1, q_1, t) \overset{*}{\vdash} (\lambda, q_1, \lambda).$$

(Example cont'd)

By the first part of a previous Theorem, $t = t'ut''$, where

$$\begin{aligned} (z_1, q_1, t') &\models^* (\lambda, q_1, \lambda), \\ (z_0, q_1, u) &\models^* (\lambda, q_1, \lambda), \\ (z_1, q_1, t'') &\models^* (\lambda, q_1, \lambda). \end{aligned}$$

Note that if z_1 is at the top of the pushdown store, then the single way to eliminate this symbol is to have $t' = a$; similarly, $t'' = a$. By the inductive hypothesis, $u = zz^R$, so $t = azz^R a = az(az)^R$. A similar argument can be made when $\mathcal{M}$ chooses the pair $(z_2 z_0 z_2, q_1) \in \delta(z_0, q_1, \lambda)$. Thus, $L(\mathcal{M}) = \{xx^R \mid x \in \{a, b\}^*\}$.