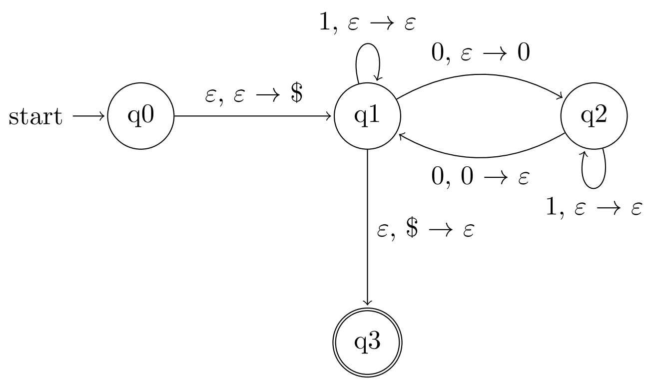
Pushdown Automata

Jan C. Bierowiec

The language $\{w \in \{0,1\}^* \mid \text{the number of 0s in } w \text{ is even}\}$.

1 State Diagram

Below is the state diagram of the PDA that accepts strings with an even number of 0s:



2 Formal Description

The PDA can be described as a 7-tuple:

$$M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$$

where:

$Q = \{q_0, q_1, q_2, q_3\}$	the set of states
$\Sigma = \{0, 1\}$	the input alphabet
$\Gamma = \{0, \$\}$	the stack alphabet
q_0	the start state
$Z_0 = \$$	initial stack symbol
$F = \{q_3\}$	the set of accepting states
δ	the transition function:

δ	ε	0	1
q_0	$(q_1,\ \varepsilon,\ \$)$	$\emptyset$	$\emptyset$
q_1	$(q_3,\ \$,\ \varepsilon)$	$(q_2,\ \varepsilon,\ 0)$	$(q_1,\ \varepsilon,\ \varepsilon)$
q_2	$\emptyset$	$(q_1,\ 0,\ \varepsilon)$	$(q_2,\ \varepsilon,\ \varepsilon)$

How the PDA Works

The PDA begins in state q_0, where it performs an ε-transition to q_1 and pushes the stack-bottom marker \$ onto the stack. From q_1, reading a 1 causes the PDA to stay in q_1 without modifying the stack. Reading a 0 causes the PDA to push a 0 onto the stack and transition to q_2. In q_2, reading a 1 again causes no stack change, but reading a 0 causes the PDA to pop a 0 from the stack and return to q_1. Thus, every 0 causes the PDA to toggle between q_1 and q_2, with the stack used to match 0s in pairs. Once the input is fully consumed and the stack contains only \$, the PDA performs an ε-transition from q_1 to the accepting state q_3 by popping the \$.

The PDA accepts a string if:

- it ends in state q_3,
- the stack is empty (i.e., \$ has been popped),
- and the number of 0s in the input is even.

Example Strings and Traces

Accepted Strings: ε, `1`, `11`, `0011`, `1010`
Rejected Strings: `0`, `10`, `011`, `000`

Trace for `1010`:

$$q_0 \xrightarrow{\varepsilon,\ \varepsilon \to \$} q_1 \xrightarrow{1,\ \varepsilon \to \varepsilon} q_1 \xrightarrow{0,\ \varepsilon \to 0} q_2 \xrightarrow{1,\ \varepsilon \to \varepsilon} q_2 \xrightarrow{0,\ 0 \to \varepsilon} q_1 \xrightarrow{\varepsilon,\ \$ \to \varepsilon} q_3 \quad \text{Accepted}$$

3 Context-Free Grammar (CFG)

The CFG that generates the same language (even number of 0s) is defined as:

$$G = (V, \Sigma, R, S)$$

where:

$V = \{S, A\}$	the set of variables
$\Sigma = \{0, 1\}$	the set of terminal symbols
S	the start symbol
R	the set of production rules

Production rules:

$$S \to 1S \mid 0A \mid \varepsilon$$
$$A \to 1A \mid 0S$$

This CFG generates all strings over $\{0, 1\}^*$ with an even number of 0s.

4 Regular Expression

Since the language is regular, it can also be described by the following regular expression:

$$(1^*(01^*01^*)^*)^*$$

This expression generates all strings with pairs of 0s, possibly separated by 1s.