



planetmath.org

Math for the people, by the people.

## non-deterministic pushdown automaton

Canonical name	NondeterministicPushdownAutomaton
Date of creation	2013-03-22 12:26:49
Last modified on	2013-03-22 12:26:49
Owner	Henry (455)
Last modified by	Henry (455)
Numerical id	14
Author	Henry (455)
Entry type	Definition
Classification	msc 03D10
Classification	msc 68Q42
Classification	msc 68Q05
Synonym	pushdown automaton
Synonym	PDA
Synonym	NPDA
Synonym	initial stack symbol
Related topic	ContextFreeLanguage
Related topic	NonDeterministicFiniteAutomaton
Related topic	Automaton
Defines	state diagram
Defines	stack alphabet
Defines	start stack symbol

## Definition

A *non-deterministic pushdown automaton* (NPDA), or just *pushdown automaton* (PDA) is a variation on the idea of a non-deterministic finite automaton (NFA). Unlike an NFA, a PDA is associated with a stack (hence the name *pushdown*). The transition function must also take into account the “state” of the stack.

Formally defined, a pushdown automaton  $M$  is a 7-tuple  $M = (Q, \Sigma, \Gamma, T, q_0, \perp, F)$ , where  $Q, \Sigma, q_0$ , and  $F$ , like those in an NFA, are the set of states, the input alphabet, the start state, and the set of final states respectively.  $\Gamma$  is the *stack alphabet*, specifying the set of symbols that can be pushed onto the stack.  $\Gamma$  is not necessarily disjoint from  $\Sigma$ .  $\perp$  is an element of  $\Gamma$  called the *start stack symbol*. The transition function is

$$T : Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \rightarrow \mathcal{P}(Q \times \Gamma^*).$$

## How It Works

To see how the computing machine  $M$  works, first imagine  $M$  with the following features:

1. a finite set  $Q$  of internal states,
2. a horizontal tape of cells each containing an input symbol of  $\Sigma$ ,
3. a tape reader that reads at most one tape cell in any given internal state, and
4. a vertical stack of cells storing symbols of  $\Gamma$ .

Now, given that  $M$  is in state  $p$ , with symbol  $A$  on top of the stack, and tape reader pointing at a tape cell containing symbol  $a$ , it may do one of the following:

- if  $T(p, a, A) \neq \emptyset$ , then it
  1. “pops”  $A$  off the stack,
  2. “pushes” word  $A_1 \cdots A_n$  onto the stack, by starting with symbol  $A_n$ , and ending with symbol  $A_1$ ,
  3. “consumes”  $a$  by moving the tape reader to the right of the cell containing  $a$ , and

4. enters state  $q$ ,

provided that  $(q, A_1 \cdots A_n) \in T(p, a, A)$ ; if  $T(p, a, A) = \emptyset$ , then  $M$  does nothing.

- if  $T(p, \lambda, A) \neq \emptyset$ , then, without reading  $a$ , it
  1. “pops”  $A$  off the stack,
  2. “pushes” word  $A_1 \cdots A_n$  onto the stack, and
  3. enters state  $q$ ,

as long as  $(q, A_1 \cdots A_n) \in T(p, \lambda, A)$ ; if  $T(p, \lambda, A) = \emptyset$ , then  $M$  does nothing.

If  $(q, \lambda) \in T(p, a, A)$ , then  $A$  gets popped off, and nothing gets pushed onto the stack.

### Modes of Acceptance

A PDA is a language acceptor. We describe how words are accepted by a PDA  $M$ . First, we start with configurations.

A configuration of  $M$  is an element of  $Q \times \Sigma^* \times \Gamma^*$ . For any word  $u$ , the configuration  $(q_0, u, \perp)$  is called the *start configuration* of  $u$ . A binary relation  $\vdash$  on the set of configurations is defined as follows: if  $(p, u, \alpha)$  and  $(q, v, \beta)$  are configurations of  $M$ , then

$$(p, u, \alpha) \vdash (q, v, \beta)$$

provided that  $\alpha = A\gamma$  and  $\beta = B_1 \cdots B_n\gamma$ , for some  $A, B_1, \dots, B_n \in \Gamma$ , and

- either  $u = av$ , and  $(q, B_1 \cdots B_n) \in T(p, a, A)$ ,
- or  $u = v$ , and  $(q, B_1 \cdots B_n) \in T(p, \lambda, A)$ .

Now, take the reflexive transitive closure  $\vdash^*$  of  $\vdash$ . When  $(p, u, \alpha) \vdash^* (q, v, \beta)$ , we say that  $v$  is derivable from  $u$ . A word  $u \in \Sigma^*$  is said to be

- accepted on final state by  $M$  if  $(q_0, u, \perp) \vdash^* (q, \lambda, \alpha)$  for some final state  $q \in F$ ,
- accepted on empty stack by  $M$  if  $(q_0, u, \perp) \vdash^* (q, \lambda, \lambda)$ ,
- accepted on final state and empty stack by  $M$  if  $(q_0, u, \perp) \vdash^* (q, \lambda, \lambda)$  for some  $q \in F$ .

## Languages Accepted by a PDA

Given a mode of acceptance, the set of words accepted by  $M$  is called the language accepted by  $M$  based on that mode of acceptance. Given a PDA  $M$ , there are three languages accepted by  $M$ , corresponding to the three acceptance modes above.

It turns out that three modes of acceptance are *equivalent*, in the following sense: if a language  $L$  is accepted by  $M$  on one acceptance mode, there are PDA  $M_1$  and  $M_2$  that accept  $L$  in the other two acceptance modes.

In general, unless otherwise stated, the language  $L(M)$  accepted by a PDA  $M$  stands for the language accepted by  $M$  on final state.

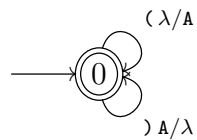
### Remarks.

1. Two PDAs are said to be equivalent if they accept the same language. It can be shown that any PDA is equivalent to a PDA where  $T(p, \lambda, A) = \emptyset$  for all  $p \in F$  and  $A \in \Gamma$  (called a  $\lambda$ -free PDA).
2. One of the main reasons for studying PDA is: the notion of a PDA is equivalent to the notion of a context-free grammar. This means that, every language accepted by a PDA is context-free, and every context-free language is accepted by some PDA.

## Representation by State Diagrams

Like an NDFA, a PDA can be presented visually as a directed graph, called a *state diagram*. Instead of simply labelling edges representing transitions with the leading symbol, two additional symbols are added, representing what symbol must be matched and removed from the top of the stack (or  $\lambda$  if none) and what symbol should be pushed onto the stack (or  $\lambda$  if none). For instance, the notation  $a \ A/B$  for an edge label indicates that  $a$  must be the first symbol in the remaining input string and  $A$  must be the symbol at the top of the stack for this transition to occur, and after the transition,  $A$  is replaced by  $B$  at the top of the stack. If the label had been  $a \ \lambda/B$ , then the symbol at the top of the stack would not matter (the stack could even be empty), and  $B$  would be pushed on top of the stack during the transition. If the label had been  $a \ A/\lambda$ ,  $A$  would be popped from the stack and nothing would replace it during the transition.

For example, consider the alphabet  $\Sigma := \{ (, ) \}$ . Let us define a context-free language  $L$  that consists of strings where the parentheses are fully balanced. If we define  $\Gamma := \{ A \}$ , then a PDA for accepting such strings is:



Another simple example is a PDA to accept binary palindromes (that is,  $\{w \in \{0, 1\}^* \mid w = w^R\}$ ).

