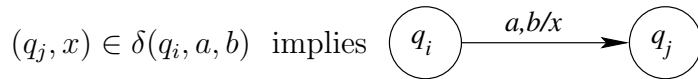


NPDA Transition Graphs

To understand NPDAs, it is useful to have a visual representation. This document introduces *NPDA transition graphs*. This type of graph extends the notation for DFA/NFA transition graphs to include notation for stack operations. This notation is borrowed from another book: J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, 2nd Edition, Addison-Wesley, 2001.

In a DFA/NFA transition graph, there is an edge from vertex q_i to vertex q_j with label a if and only if the finite acceptor can make a transition from state q_i to state q_j while reading input symbol a . [In a NFA, if the label is λ , no input symbol is read.] For a NPDA transition, we also have to know what symbol is popped off the stack and what symbols, if any, are pushed on the stack. The graph notation for an initial state and final states are the same in NPDA transition graphs.

In a NPDA transition graph, there is an edge from vertex q_i to vertex q_j with label $a, b/x$ if and only if the NPDA can make a transition from state q_i to state q_j while reading input symbol a , popping stack symbol b , and pushing a string x on the stack. In the δ transition function notation, this corresponds to:

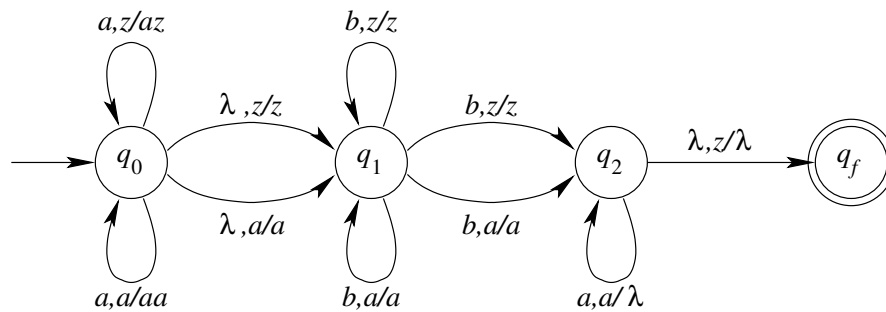


Similar to NFAs, a label that begins with λ indicates that no input symbol is read.

For example, the following NPDA accepts the language $\{a^k b^n a^k : k \geq 0, n \geq 1\}$.

$Q = \{q_0, q_1, q_2, q_f\}$	$\delta(q_0, a, z) = \{(q_0, az)\}$
$\Sigma = \{a, b\}$	$\delta(q_0, a, a) = \{(q_0, aa)\}$
$\Gamma = \{z, a, b\}$	$\delta(q_0, \lambda, z) = \{(q_1, z)\}$
initial state = q_0	$\delta(q_0, \lambda, a) = \{(q_1, a)\}$
stack start symbol = z	$\delta(q_1, b, z) = \{(q_1, z), (q_2, z)\}$
$F = \{q_f\}$	$\delta(q_1, b, a) = \{(q_1, a), (q_2, a)\}$
	$\delta(q_2, a, a) = \{(q_2, \lambda)\}$
	$\delta(q_2, \lambda, z) = \{(q_f, \lambda)\}$

Here is the equivalent NPDA transition graph:



Hopefully, the NPDA transition graph makes it clearer what each state and transition does. q_0 is used to read and push any number of a 's. The transition from q_0 to q_1 can be made at any time. q_1 is used to read any number of b 's while leaving the stack unchanged. The transition from q_1 to q_2 ensures that at least one b is read. q_2 is used to read and pop any number of a 's. The transition from q_2 to q_f is possible only if all the a 's have been popped off the stack.