The Nondeterministic Push-Down Automaton Simulator Program

The Non-deterministic Push-Down Automaton Simulator program simulates the execution of an NPDA or PDA by allowing the generation and running of small "programs". Instructions to the PDA, which are entered as quintuples of symbols, can be created and edited using a built-in Editor feature. The user can initialize the tape to any input string of non-blank symbols and set the starting state for the PDA. The user can execute the PDA quintuples until a "halt" state is obtained or the user can "single step" the PDA one quintuple instruction at a time. If a PDA configuration permits multiple actions (non-determinism), a window pops open allow the user to select the course of action to take.

The NPDA Simulator runs under in MS-DOS command window.

When the NPDA simulator is executed, after the initial title screen the user will be presented with a four-window screen where each window, except the bottom (command) window, is initially empty.

<table>
<thead>
<tr>
<th>PDA Input Tape and Stack</th>
</tr>
</thead>
<tbody>
<tr>
<td>.00110010..................</td>
</tr>
<tr>
<td>B</td>
</tr>
<tr>
<td>0011z</td>
</tr>
<tr>
<td>Head Position = 5 Iteration = 5 Quintuple: (B,ε,0,0,C)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PDA Quintuples</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A,0,z,0z,B) –</td>
</tr>
<tr>
<td>(A,1,z,1z,B) –</td>
</tr>
<tr>
<td>(B,0,0,00,B) –</td>
</tr>
<tr>
<td>(B,1,1,11,B) –</td>
</tr>
<tr>
<td>(B,0,1,01,B) –</td>
</tr>
<tr>
<td>(B,1,0,10,B) –</td>
</tr>
<tr>
<td>(B,ε,1,1,C) –</td>
</tr>
<tr>
<td>(B,ε,0,0,C) –</td>
</tr>
<tr>
<td>(C,0,0,ε,C) –</td>
</tr>
</tbody>
</table>

| Editor Go Quit Reset Stack State Tape 1 Step Single Step PDA |

Description of the Four Windows:

**Top Window (PDA Input Tape and Stack):** This window has two parts. The upper part displays the current configuration of the PDA by showing the current contents of the Input Tape, the position of the Read Head (denoted by \( \cdot \) symbol above) and the Current State (character beneath the Read Head). The lower part displays the contents of the stack. The top of the stack is the left-most character. \( z \) is the special stack start symbol.

Tape symbols must be non-blank characters. Dot characters are used to indicate blanks which are not part of the tape input.

**Second Window (Status):** Displays the current status of the PDA; that is, Position of the Read Head, the number of steps executed by the PDA (Iteration) and the Quintuple to be executed next. If the PDA is halted or interrupted, the words Running, HALT - No, HALT - Yes, or INTERRUPT will also display. Quintuples that display a '?' in the next stack/next state fields indicate that no quintuple exits with the current state/current input symbol/current stack values. This will cause the PDA to halt.

**Third Window (PDA Quintuples):** This is a small editor which allows the user to create and edit the quintuples used to "program" the PDA. The Quintuples window is accessed using the Editor option as discussed below.
Bottom Window (Command): This displays the current user options. If the Editor option is chosen, a second menu displays which lists the options available for use with the Editor. Selections may be made by either using the cursor keys to highlight the option and hitting (Enter) or by hitting the corresponding letter key.

Main Menu User Options:

Editor: Activates the Editor allowing the creation/editing of PDA quintuples in the Quintuples window. A second list of options appears.

Go: Executes the PDA using the quintuples "program" currently displayed in the PDA Quintuples window. The input tape, the start and accept states, and the stack should be initialized or reset first. Execution may be interrupted by hitting the (Esc) key.

Quit: Quits the program.

Reset Stack: Resets the stack to contain the special Stack Start Symbol 'z'. This symbol is used to indicate the bottom of the stack.

State: Allows the user to set the Initial and Accept States.

Tape: Allows user to edit the Input Tape and position the Read Head. Be sure to reposition the Read Head at the beginning of the Input Tape before executing a program. Use the State option to initialize the current state.

1 Step: Same as Go except allows user to use the space bar to single step through the execution of PDA quintuples.

The Editor:

The Editor allows the user to create and edit PDA quintuples. A different set of options is displayed in the bottom window when this option is chosen.

Edit Text: Starts the Editor and allows the User to edit the current contents of the Quintuples window. Hitting (Esc) or the [F10] keys will return the user to the Command window.

Include File: Allows the User to include the contents of any ASCII text file at the current location of the cursor in the Quintuples window area. User is prompted for the file name.

Main Menu: Returns the User to the previous menu. When the User exits the Editor option, the current contents of the Quintuples window become the set of quintuples used by the PDA. Any "previous" quintuples are overwritten. Some checking is done on the format of the quintuples.

New Text: Clears current contents of and puts User into the Quintuples window.

Read File: Clears current contents of Quintuples window and reads contents of a file into Quintuples window. Useful if Users wishes to recall previously saved quintuples.

Save File: Allows user to save current contents of Quintuples window to file.
**PDA Quintuples:**

The format of a quintuple must be strictly followed. Each quintuple must be enclosed in parentheses and use commas to separate the fields. Each quintuple consists of five fields separated by commas. The fields are

- current state (one or two non-blank characters)
- current input symbol
- top of stack symbol
- symbol(s) to be pushed on stack (up to three non-blank characters)
- next state (one or two non-blank characters)

A quintuple cause a transition if the PDA's current state, current input symbol, and top of stack match the first three entries in the quintuple. When a match occurs

- the current input symbol is consumed (the Read Head moves to the right)
- the top of the stack is popped (and thrown away)
- the symbol(s) to be pushed on stack (4th component of quintuple) are pushed onto the stack
- the next state (5th component of the quintuple) becomes the new current state.

**Example:** The quintuple

\[(q0, a, z, AAz, q1)\]

will cause a transition if the current state is \(q0\), current tape symbol is \(a\) and the top of the stack is \(z\). The net result to the PDA is to push \(AA\) onto the stack (actually the \(z\) is popped off the stack and then \(z\) followed by \(AA\) is pushed onto the stack) and go to state \(q1\).

**Example:** The quintuple

\[(q2, b, A, B, q3)\]

will pop the \(A\) off the stack, push \(B\) onto the stack and go to state \(q3\).

**\(\varepsilon\)-Transitions:**

An \(\varepsilon\) (epsilon) symbol indicates no input is consumed when it appears as the 2nd component in a quintuple or no symbol is pushed onto the stack when it appear as the 4th symbol in a quintuple. The latter is used to pop the stack. Epsilons can appear nowhere else in a quintuple.

**Example:** The quintuple

\[(q3, \varepsilon, A, A, q4)\]

consumes no input so the Read Head does NOT move to the right. If the current state is \(q3\) and the top of the stack is \(A\), then ignoring the current input symbol, keep \(A\) on top of the stack (pop \(A\) and push \(A\)) and go to state \(q4\).

**Example** The quintuple

\[(q5, b, B, \varepsilon, q5)\]

pops the stack. If the current state is \(q5\) and the current tape symbol is \(b\) and the top of the stack is \(B\), move the Read Head to the right (consume input), pop the stack (the stack is always popped but \(\varepsilon\) says do not push anything) and stay in state \(q5\).
Non-Determinism:

If there are multiple matches of quintuples with the current state, current tape symbol and top of stack, a window will pop open displaying all such quintuples. The user then selects which quintuple to execute.

Additional Notes on NPDA Quintuples

States both current and next are restricted to single letters, single digits or pairs of letters and/or digits (e.g. A9, 1A, or 23).

The current input symbol can be any non-blank character including the special character $\varepsilon$. The $\varepsilon$ is generated in the Editor using Alt/E or Alt/L denotes an epsilon (or lambda) transition which consumes no input.

The top of stack symbol can be any non-blank character. A special Stack Start Symbol ‘z’ is used to indicate the bottom of the stack. Before starting a NPDA or PDA program, the stack should be initialized (see Reset Stack option) to hold this character.

The symbol(s) to push on stack can be any single, pair or triple of non-blank characters which replaces the current top of stack (the stack is always popped). The $\varepsilon$ character (Alt/E or Alt/L) is used to indicate no character is pushed onto the stack - "a pure pop” operation. Multiple characters are pushed right to left (so if ABC is pushed on the stack, A is the top character).

Since everything to the right of a Quintuple is ignored, descriptive comments may be attached to quintuples.

To Accept or Not Accept: Halt - Yes versus Halt - No:

The NPDA will halt on a configuration (current state, input symbol, stack symbol) for which it has no quintuple. A Halt -Yes which indicates that the NPDA accepts the input only occurs if the NPDA is at the end of the input tape and current state is the designated Accept State (see State option under the Main Menu User Options). The contents of the stack is irrelevant to the determination of an Accept State. Otherwise Halt - No is obtained.

Example PDA and NPDA Programs

A PDA Program: Detecting strings of the form $0^n1^n$ for $n > 0$.

The following PDA quintuples detect strings where one or more 0's are followed by one or more 1’s. A is the Start State; C is the Accept State.

\[(A, 0, z, 0z, A)\] - push 0 on stack
\[(A, 0, 0, 00, A)\]
\[(A, 1, 0, \varepsilon, B)\] - 1 detected; pop stack ($\varepsilon$)
\[(B, 1, 0, \varepsilon, B)\]
\[(B, \varepsilon, z, z, C)\] - end of string and z on stack; accept

In state A, each 0 on the input tape is pushed onto the stack. When a 1 is detected, a zero if popped off the stack. If the number of 0's and 1's is the same, the stack will be empty (z is end-of-stack character) at the same time we reach the end of the input tape.
An NPDA Program: Detecting Palindromes

A palindrome of 0's and 1's is any string of 0's and 1's that reads the same forward and backwards. Palindromes can be even (e.g. 101101) or odd (e.g. 10101) depending whether the number of symbols is even or odd. Non-determinism is needed to detect palindromes since we don't know where the first half ends and the mirror second half begins.

(A,1,z,1z,B)    Palindrome Detector
(A,0,z,0z,B)    A - Start State
(B,0,0,00,B)    push string on stack
(B,0,1,01,B)    
(B,1,0,10,B)    
(B,1,1,11,B)    
(B,ε,1,1,C)    change to state C - even palindrome
(B,ε,0,0,C)    
(B,0,0,0,C)    change to state C - odd palindrome
(B,0,1,1,C)    
(B,1,0,0,C)    
(B,1,1,1,C)    
(C,0,0,ε,C)    compare input with stack
(C,1,1,ε,C)    
(C,ε,z,z,D)    D - Accept State

The above NPDA program begins by pushing each 0 and 1 from the input tape onto the stack. At some point it decides that the first half of the string has been pushed onto the stack and switches to state C where the second half of the input tape is compare to the contents of the stack. This is where the non-determinism comes in. In addition even and odd palindromes must be handled differently. Even palindromes are handled by quintuples which consume no input. Odd palindrome are handled by quintuples which do not push the "middle" tape symbol on the stack. From state C the NPDA compares each tape symbol with the top of the stack, popping the stack if they match. At the end of the input tape if the stack start symbol is encountered, the NPDA goes to state D, the accept state.

Recall that a NPDA accepts its input if there is some sequence of configurations that ends in an Accept state. This means that when you execute the above NPDA program you must correctly choose when to advance to state C.

Executing a NDPA or PDA Program

To execute a NPDA or PDA program you need a set of quintuples, a non-empty input tape, the stack initialized to contain only the special stack start symbol, and Start and Accept states.

1. From the Main Menu select the Editor option and enter your NPDA or PDA quintuples into the Edit window. Exit the Editor and return to the Main Menu. The NPDA program uses the quintuples from the last edit session.

2. From the Main Menu select Tape and enter the input tape. When done use the arrow keys to reposition the Read Head at the beginning of the input (on the first non-blank symbol).

3. From the Main Menu select Reset Stack to initialize the stack so that only the special stack start symbol ("z") is on the stack.

4. From the Main Menu select State and at the prompts enter the Start State and the Accept State.

5. From the Main Menu, select either Go or 1 Step.
Notes:

If a PDA configuration permits two or more actions (non-determinism), a window will pop up displaying all possible quintuples for that PDA configuration. Use the up and down arrow keys to highlight a quintuple then hit (Enter) to execute select it.

States are restricted to letters and digits. States may be one or two characters long. Tape symbols are all non-blank single characters. Commas are not allowed. State and Tape symbols are case sensitive.

Up to three characters may be pushed onto the stack by one transition.

Avoid the using '?' as a tape or state symbol since it is used to indicate that a quintuple with a given current state/tape symbol pair was not found. Do no use blanks as tape symbols as a blank is used to indicate the end of the input tape.

A quintuple must have the correct format. If the syntax of a quintuple is incorrect, it is ignored.

The length of the input is restricted to 1000 characters. The Stack is restricted to 75 characters.

An empty stack automatically halts the NPDA.

Hitting the [F1] key will display a Help window.

Hitting (Esc) will interrupt a running NPDA program. An infinite loop might occur with a quintuple that consumes no input and does not change the stack symbol.