**Requirements Specification**

**For the Generations of The Joker Pushdown Automaton Application**



**Cpts: 422, Software Engineering Principles II**

**Due: 10/22/19**

**Eric Engel**

**Trevor Surface**

**Micheal J. Roper**

**Bobby Thompson**

**Table of Contents**

[Introduction of Application 4](#_Toc22403302)

[Background of a Pushdown Automaton 5](#_Toc22403303)

[Overview of Purpose and Development 7](#_Toc22403304)

[Environment 7](#_Toc22403305)

[1. Input and Output Devices 8](#_Toc22403306)

[Turing Machine Definition File 8](#_Toc22403307)

[Input String File 10](#_Toc22403308)

[Operation 11](#_Toc22403309)

[Invocation 11](#_Toc22403310)

[Command Line 11](#_Toc22403311)

[Configuration Settings 12](#_Toc22403312)

[Opening Turing Machine 12](#_Toc22403313)

[Commands 13](#_Toc22403314)

[Help User 13](#_Toc22403315)

[Show Status 13](#_Toc22403316)

[View Turing Machine 14](#_Toc22403317)

[List Input String 14](#_Toc22403318)

[Insert Input String 14](#_Toc22403319)

[Delete Input String 14](#_Toc22403320)

[Set Transitions 14](#_Toc22403321)

[Truncate Instantaneous Description 15](#_Toc22403322)

[Run Turing Machine 15](#_Toc22403323)

[Quit Turing Machine 15](#_Toc22403324)

[Exit Application 15](#_Toc22403325)

[Termination 16](#_Toc22403326)

[Closing Turing Machine 16](#_Toc22403327)

[References: 17](#_Toc22403328)

[Appendix: 17](#_Toc22403329)

# 

**List of Figures**

Figure # Title Page #

1 Generation of The Joker Pushdown Automaton Example 7

2 Generation of The Joker Context Diagram 8

**Document Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Revision Number | Revision Date | Description | Rationale |
| 4 | 10/22/2019 | Final Revision | Final revision after grading |
| 3 | 10/20/2019 | Final Review | Review document for final revisions being made |
| 2 | 10/19/2019 | Additional Details | Adding details to the document |
| 1 | 10/17/2019 | Initial Setup of documentation | First draft getting the major points on the document. |
|  |  |  |  |
|  |  |  |  |

# Introduction of Application

The purpose of this document is to describe and layout all the specific requirements for the Generations of The Joker pushdown automaton application. The intended audience for this application is for the users to have knowledge and understanding within computer theory, automata theory and pushdown automaton applications.

This document will layout the foundation as to what a pushdown automation is, then dive into the purpose of this application and specs for development. Next will be the environment in which will be used along with the layout needed to run this application. Finally, will be the details of how to start the application and once the application is running how to then properly use the commands of the application. Along with giving detailed description as to what every command will show and prompted for from the user.



Picture 1

# Background of a Pushdown Automaton

The pushdown automaton (PDA) is a type of an automaton that employs a stack. Automaton or Automata theory is the study of abstract machines and automata as well as the computational problems that can be solved using them. Pushdown automatons are used in theories about what can be computed by machines. They are more capable than finite-state machines but less capable than Turing machines. The term “pushdown” in PDA refers to the stack that is being implemented within the design of the PDA

The pushdown automaton can be formally defined by **M = {Q, Σ, Γ, δ, q0, Z0, F}.**

* **Q** is a finite, set of states;
* **Σ** is the set of input alphabet symbols, that is, the set of symbols allowed to appear in the initial tape contents;
* **Γ** is a finite, set of stack alphabet symbols;
* **δ** is a partial function called the transition function, where L is left shift, R is right shift. If is not defined on the current state and the current tape symbol, then the machine either halts or crashes.
  + Displayed in this format: ***Q* × (Σ ∪ {λ}) × Γ → ℘(*Q* × Γ\*)**
* **q0**is the initial state, is a state in **Q**;
* **Z0** is the start symbol on the stack,
* **F** is the set of final states (accepting states), is a state(s) in **Q**.

Another character is used in a pushdown automaton is the empty string which is indicated by the symbol **λ**. Also, a pushdown automaton can only run on an input string consisting of symbols from the input alphabet. All other symbols are not allowed on any input strings and will make that string invalid if they are found. Instantaneous Description (ID) describes the current configuration of a pushdown automaton at run-time. A pushdown automaton has 0 or more final states which cause it to halt when entered, accepting original input string. If the pushdown automaton does not enter a halt state, the pushdown automaton will crash cause resulting in rejection of original input string, or an infinite loop.

In Figure 2-1 is an example of a pushdown automaton with a formal definition of:

* **Q** = {S0, S1, S2} (states);
* **Σ** = {a, b} (input alphabet);
* **Γ** = {A, Z} (finite stack alphabet);
* **q0 =** {S0} (initial state);
* **Z0** ={Z} (initial stack symbol);
* **F =** {S2} (final state(s));
* **δ** = transition functions:
  + δ(s0, a, X) = {(s0, XX), (s1, X)}
  + δ(s0, a, Y) = {(s0, XY), (s1, Y)}
  + δ(s0, a, Z) = {(s0, XZ), (s1, Z)}
  + δ(s0, b, X) = {(s0, YX), (s1, X)}
  + δ(s0, b, Y) = {(s0, YY), (s1, Y)}
  + δ(s0, b, Z) = {(s0, YZ), (s1, Z)}
  + δ(s1, a, X) = {(s1, λ)}
  + δ(s1, b, Y) = {(s1, λ)}
  + δ(s1, λ, Z) = {(s2, λ)}

If we were to run this pushdown automaton on a couple examples, we could see when the string was accepted or rejected.

Example String inputs on figure 2-1 Generations of The Joker Pushdown Automaton Example.

* **λ** rejected
* aba rejected
* abaa accepted

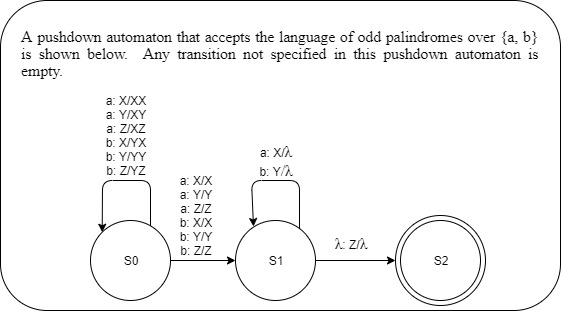


Figure 1: Generation of The Joker Pushdown Automaton Example

# Overview of Purpose and Development

The purpose of the Generations of The Joker Pushdown Automaton Application is to allow a user to interactively trace the operations of an arbitrary pushdown automaton on an input string. The development of this Generations of The Joker Application will be done with the waterfall method of gathering all requirements shown in this document. This will be a console application, on a PC running the ubuntu Linux operating system and the GNU g++ complier. The application is implemented in C++ language using the C++ standard library as the only library for this application.

User interaction with the application will be from the ubuntu command console. This will allow the user to input commands from the keyboard and receive responses back to be displayed on the console. The monitor will also display prompts for input that the user can then provide.

Context diagram of the application and what is expected of the program in general can be found in Figure 2. The diagram shows that how the user input responses from the keyboard. Both the configuration file and definition file will send their information into the pushdown automaton. While the input string file will allow file information to be sent and received from the pushdown automaton to allow for additional input strings by the user to be added to the file.

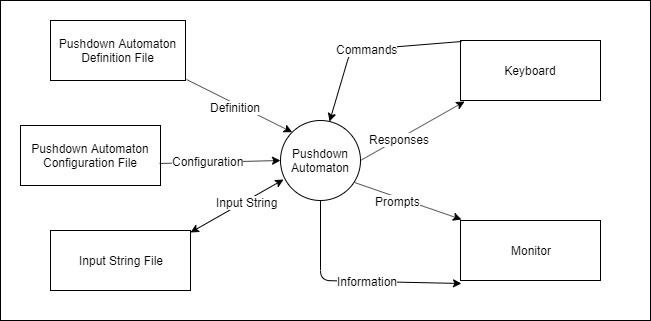


Figure 2: Generation of the Joker Context Diagram

# Environment

The input and output devices along with the text files that will be used for the Generations of the Joker application are described in detail further in this section.

## Input and Output Devices

The input and output devices for this application are the keyboard for the input and Monitor for reading output. The Monitor will be for reading information being displayed while seeing the prompts and entering the proper commands from the keyboard.

## Configuration File

The configuration file will be formatted according to the following specs, if the file is not formatted in this way the Generations of The Joker application will provide an appropriate error message(s) to the monitor for the user and exit the application. Generations of The Joker application will not use Erroneous files and will require the user to correct them with a text editor. Configuration file contain 2 settings, the first will set the maximum number of transitions to perform at a time during operation of application, default is 1. The second configuration setting is the maximum number of cells to left and right of tape to display in an instantaneous description, default is 32.

## Pushdown Automaton Definition File

The pushdown automaton definition file will be formatted according to the following specs, if the file is not formatted in this way the Generations of The Joker application will provide an appropriate error message(s) to the monitor for the user and exit the application. Generations of The Joker application will not use Erroneous files and will require the user to correct them with a text editor.

The definition file name will have “.def” appended to the end of the file. The file will contain 7 keywords in the proper order. Keywords may appear in upper or lower case or in any combination of upper and lower case. Some Keywords will contain an underscore within the name and each keyword will end with a colon. All 7 keywords must be included in the file and be in the proper order and format shown: **STATES: INPUT\_ALPHABET**: **TAPE\_ALPHABET**: **TRANSITION\_FUNCTION: INITIAL\_STATE: BLANK\_CHARACTER: FINAL\_STATES:**

Any combination of white space which includes spaces, tabs, and new lines in the file is acceptable. Once a keyword has been read everything in front of that keyword will be considered part of that keyword’s parsing definition until next keyword is read.

All text in this file up to the first keyword (states:) will be considered description of the pushdown automaton definition file and will be ignored in parsing the file.

**States:**

Each state is named as a string of upper- or lower-case letters, digits, or underscore. Each state name is case sensitive and must be unique.

**Input\_Alphabet: & Tape\_Alphabet:**

Input alphabet and Tape Alphabet will consist of only printable characters from the ascii character set except for \ [] < >. White space is also excluded from the alphabet. Every character in the alphabet must be unique. Every character in the input alphabet must be in the tape alphabet. Blank character must be in the tape alphabet, but not in the input alphabet and must be single character.

* **\** (**λ)** is the notation of the empty string.
* **[]** are for displaying the state within the tape.
* **< >** are for displaying the tape head with truncation.

**Transition\_Function:**

Any number of transitions may be defined in the transition function. Each transition must use states in set of states and characters in the tape alphabet. Transitions must also define direction to move tape head left or right using the character L or R in upper or lower case. At most one transition from a given state on a given tape character may be defined, since the Turing machine is deterministic. If a state is defined to be a final state, no transitions from that state will be allowed.

**Initial\_State:**

Must only be one state be in the set of states.

**Blank\_Character:**

Must be a character in tape alphabet and only one character allowed.

**Final\_States:**

Any state in the set of states may be defined to be a final state, each final state must be

unique.

## Input String File

Input string file contains input strings from the input alphabet of the pushdown automaton definition file one per line. Input string file has same name as the pushdown automaton definition file except for extension of “.str”. Empty string(**λ**)is a specified by **\** appearing on a line by itself. No white space in allowed in the file. If any line is empty or contains a character not in the input alphabet, Generations of The Joker application will discard the line, duplicate input strings will also be discarded. There is no limit of length of an input string or number of input strings. Each input string is evaluated on a line by line basis. An example of a properly formatted input string file will be shown in the figure 4-2.

# Operation

During the operation of the pushdown automaton, the Generations of The Joker application will always keep track of 1 of 4 status of the current pushdown automaton. First status, pushdown automaton has not been run on an input string during the session. Second status, pushdown automaton is currently running on an input string. Third status, pushdown automaton has completed operation on an input string during the session. Finally the fourth status being closed status of the PDA.

## Invocation

### Command Line

All operations from the user will be done through the command line when prompted to do so. No message will be displayed on a black response. Feedback will be provided on any other response given. Once a pushdown automaton is successfully loaded, commands are available to perform on pushdown automaton until user exits the application. Operations are performed in application through single character commands enter by user in upper or lower case in response to prompts “Command: “. The command structure is flat, with all commands appearing at the top level rather then grouped into menus according to topics.

### **Configuration File Settings**

Two Configuration settings will be set to a default at beginning of operation but may be changed during operation or from input of a file. Changes are not saved when application is terminated. Configuration setting one is maximum number of transitions to perform at a time during operation of application, default is 1. Configuration setting two is maximum number of cells to left and right of tape to display in an instantaneous description, default is 32. The tape head is considered part of right side of the head tape when displayed.

Generations of The Joker application can perform operations on pushdown automaton when a valid definition file is read. File is specified when application is executed through a command line argument as seen in figure 5-1. In specifying file, user may provide full or relative path, without extension of “.def”. Name of definition file is formed from name with extension of “.def” appended. Name of input string file is formed from name with extension of “.str” appended. If other than 1 command line argument is provided, an application usage message is displayed, and execution is terminated. If Turing machine definition file does not contain a valid Turing machine definition, appropriate error messages are provided to user on monitor and execution is terminated. If Turing machine is successfully loaded from definition file, any description of Turing machine found in file is saved for viewing the Turing machine by user. If input string file exists, all valid input strings from the file are loaded into list of input strings. Invalid input strings result an error message and the string to be discarded. If input string file does not exist or file does not contain any valid input strings, list is initially empty.

### Opening Pushdown Automaton

The user will start the application using the proper command: “./gen\_of\_joker\_pda {PDA definition file name}”.

Before starting the application, the user will insert a PDA definition file along with an input string file (users’ choice). Once started the Generations of The Joker application will check if the PDA definition is in the proper Generations of The Joker PDA required format to be accepted. If the definition is accepted it will then be loaded into the application, if not the application will reject the definition and exit with the proper errors to the monitor for the user. During the time of loading and checking of the definition file, if the user provided an input string file this will be loaded and checked. All invalid input strings will be discarded at the time of loading. If all is successful, the user will be prompted by the Generations of The Joker application to enter a command with the command prompt appearing on the monitor.



Picture 2: “The only sensible way to live is without rules”

**Start Example:**

**$**./gen\_of\_joker\_pda pda\_file

gen\_of\_joker\_pda Loaded Successfully with PDA: pda\_file.def

gen\_of\_joker\_pda Loaded Successfully with Input String: pda\_file.str

Command:**\_**

## Commands

### Help User

Help command: {H, h}

Displays all commands of application, characters used for selection of command, and a brief description.

### Show Status

Show command: {W, w}

Displays status of application: course, author, semester, year, version of application, instructor, followed by a blank line. Then displays status of two configuration settings maximum transitions and maximum cells in ID along with their set values from the configuration file. Another blank line and finally the name of the current Turing machine without extension of “.def”. Depending on the Status that is set current turning machine more information could be displayed. If status 2 is set will display input string, total number of transitions performed so far. If status 3 is set, will display the last input string and whether it was accepted, rejected, or terminated by user. Finally, will display the total number of crashes and show number of transitions in the successful path.

### Open Pushdown Automaton

Open command: {O, o}

The open command has an implicit close built into it in the case that the PDA is already operating on an input string and/or open PDA. Quitting an operation on an input string, write input string list if needed, close PDA. Open Prompts for the name of the new PDA to be opened, if you back out at this point, any previously opened PDAs have already been closed, therefore you will currently be in the Closed state. If this command errors, shows an error message, then return to command prompt with closed PDA. However, on success, show the success command and then prompt for a command from the user.

### View Pushdown Automaton

View command: {V, v}

Displays definition of currently loaded PDA on monitor in form readable by the user. Description of PDA from definition file must be displayed on original format. Followed by the formal definition of the Turing machine **M = {Q, Σ, Γ, δ, q0, Z0, F}** in order, displayed on monitor, with as much notation from formal definition specification as possible to include curly brackets, commas, etc.

### List Input String

List command: {L, l}

Displays list of all input strings upon which the PDA may run. Each input string should appear on its own line and be numbered sequentially, starting at 1, for reference by user in other commands of applications. If list is empty, this should be indicated to the user.

### Insert Input String

Insert command: {I, i}

Allows user to enter input string from the input alphabet and append it to list of input strings at the end of list. If input string entered by user is already in the list or contains a character not in the input alphabet, an error message is displayed, and the string is discarded. No error if no input string is entered, leaving list unchanged. If **\** (empty string) is inserted into input string will be entered in an empty string.

### Display Paths

Paths command: {P, p}

Displays all the completed paths of currently loaded PDA on monitor in a form readable by the user. This will only show if the completed path through the PDA, if no paths have been completed this a message will be displayed saying that to the user.

### Run Pushdown Automaton

Run command: {R, r}

Allows user to trace operation of pushdown automaton on an input string selected from list. If pushdown automaton is not already running, user is prompted for number of input string number upon which to run on pushdown automaton. If user selects a nonexistent input string, error message is displayed, and command is terminated. No error on no number entered, terminating the command. Initial instantaneous description (ID) displayed, it is of the form α1 q α2. Each ID is numbered with total number of transitions performed up to that point. In displaying ID, q should be enclosed with **[** and **]**, and α1 and α2 may be truncated using setting for maximum number of cells to display. **<** is used to truncate α1 at left end. **>** is used to truncate α2 at right end. After displaying initial ID, if initial state is also a final state, string is accepted in 0 transitions. If newly selected input string is not accepted or pushdown automaton is already running on an input string. Transition may be preformed to trace operations of pushdown automaton until it accepts or rejects input string or user quits running the pushdown automaton. After preforming up to maximum number of transitions specified by configuration file, current ID is displayed on monitor. Number of transitions preformed may be less than maximum for this iteration of run command, if input string is accepted or rejected. If pushdown automaton rejects input string, ID displayed has caused pushdown automaton to crash, therefore should get a reject message. If pushdown automaton accepts or rejects input string, display total number of transitions performed as part of the message along with input string.

### Quit Pushdown Automaton

Quit command: {Q, q}

Terminate operation of pushdown automaton on an input string before completion. Message should be provided indicating that the pushdown automaton has not accepted or rejected input string. Includes total numbers of transitions preformed so far and input string. Error message if pushdown automaton is not running on an input string.

### Close Pushdown Automaton

Close command: {C, c}

Same as the Open Command excepts does not attempt to open a new PDA.

In the case that the PDA is already operating on an input string and/or open PDA. The close command will quit the operation of the input string, write input string list to file if needed, and close PDA. If this command errors, shows an error message, then return to command prompt with closed PDA.

### Exit Application

Exit command: {X, x}

Exits application providing no opportunity to confirm. If an input sting was inserted or deleted from list of input strings, or string was discarded at load time, entire list is written to input string file, replacing any original file. Success message or error message will be displayed as the result of the success or failure of writing the file. No changes to input string, do not attempt to write file. Only if succeeding in changing list will rewrite over file.

# References:

Documentation and Detailed information provided by Professor Neil Corrigan of CPTS 322 spring 2019

Title page picture: <https://www.amazon.com/Jokers-Batman-Movie-Actors-Poster/dp/B01N68OS5J\>

Footer picture: <https://wallpapersden.com/joker-2019-wallpaper/480x484/>

Picture 1: <https://www.fanjackets.com/blog/ultimate-joker-costumes-guide/>

Picture 2: <https://www.reddit.com/r/batman/comments/1stsv4/i_love_this_concept_art_from_arkham_origins_the/>

# Appendix:

For more information on Pushdown Automaton visit: <https://en.wikipedia.org/wiki/Pushdown_automaton>

For more information on Automata Theory visit: <https://en.wikipedia.org/wiki/Automata_theory>

For more information on C++ visit: http://www.cplusplus.com/