Automata Runner User's Guide

By

David Dahlbacka

The Automata Runner User's Guide *is an adjunct to Automata Runner, a college-level educational Windows program that supplements the teaching of Automata Theory in the classroom. The User's Guide supports the user in interactively specifying, displaying, running, and debugging certain classes of formal automata.*

Acknowledgements

Microsoft Visual Studio .Net Framework Copyright 1985-2001 by Microsoft Corporation. All rights reserved.

Adobe Acrobat Copyright 1984-2004 Adobe Systems Incorporated and its licensors. All rights reserved.

Contents

1 What is Automata Runner?

Automata Runner is a college-level educational Windows program intended to supplement the teaching of Automata Theory in the classroom. It provides an interactive environment for emulating the operation of formal automata.

Formal automata are very general mathematical entities that can model a wide variety of real systems. The simplest formal automaton, a finite state machine (FSM), can model systems as diverse as:

- A restricted set of languages, where the symbols may be sounds, characters, or arbitrary objects, such as visual or architectural elements.
- Simple physical mechanisms, such as internal combustion engines and vending machines.
- Digital logic controllers, such as field-programmable gate arrays (FPGAs).

A stack machine (SM) can model a more extensive set of languages, including most computer languages. A Turing machine (TM) is mathematically equivalent to any information processing system, while a cellular automaton (CA) is theoretically capable of modeling any physical system. Finally, a Von Neumann machine (VNM), also called a "stored-program computer," is the automaton that best models almost all of the computers in use today.

Using Automata Runner, you can define a formal automaton, display it, and run it. By providing a visual display of an otherwise abstract mathematical concept, Automata Runner teaches you about the capabilities, similarities, and differences among automata. By emulating the automaton's behavior, Automata Runner teaches you the basis for a proof by induction.

The Automata Runner documentation includes a *User's Guide* and a *Study Guide* in student and instructor versions. The *Automata Runner User's Guide* shows you how to:

- Install Automata Runner.
- Define an automaton and save it to disk.
- Select a visual display mode for its operation.
- Run the automaton and debug its behavior.
- Customize the specification and visual display.

The *Automata Runner Study Guide*, included with the default installation of Automaton Runner, includes the following material:

- Overview of Automata Theory
- Overview of Theorem-Proving Methods
- Overview of Automata Applications
- Student Exercises

- References

The passworded Instructor's version of the *Study Guide* includes the following additional material:

- Lecture notes
- Answers to Student Exercises

2 Getting Started

This section gives the information you need to install Automata Runner, troubleshoot the installation, uninstall Automata Runner, and bring up the program and its auxiliary materials.

2.1 System Requirements and Installation

Automata Runner has the following system requirements:

- Windows XP, service pack 2 or later.
- Intel Pentium III processor, 1GHz processor speed or better.
- 100MB of free disk space.
- 1GB of physical memory.
- CD-ROM drive.
- Microsoft Visual Studio .Net Framework, version 1.5. If this or a later version is not installed on your computer, the installation procedure will automatically install it.
- Adobe Acrobat, version 5.0. If this or a later version is not installed on your computer, the installation procedure will automatically install it.
- Internet access (for product registration).

You install Automata Runner as follows:

1. Close all open applications.
2. Insert the Automata Runner Install disk in CD-ROM and close the drive. The installation procedure should run automatically, but if it does not, browse to the CD-ROM drive and double-click Setup.exe.
3. Follow the prompts and enter the following information:
    - User's name.
    - Installation directory. The default directory is:

      C:\Program Files\AutomataRunner.
    - Product serial number. This is the 16-character string labeled "Product Serial Number" on the disk label and on the back of the jewel case. Keep the serial

number in a secure place, because you will need it to reinstall the software or to get help from product support.

4. When the installation procedure prompts you to insert the Study Guide Install disk, do so, and follow the prompts to enter the following information:
   - Study Guide serial number. This is the 16-character string labeled "Study Guide Serial Number" on the disk label and on the back of the jewel case.
   - If you are installing the Instructor's version of Automata Runner, the installation program will prompt you for the instructor's key included with the installation package in a sealed envelope. Keep the key in a secure place. Enter the key, and follow the prompts to enter the following information:
     - Instructor's name.
     - Instructor's password.
5. When you are done with the installation, you will see a screen stating that the installation was successful and asking if you wish to register the product. Formal registration of Automata Runner is not necessary for you to use it. However, if you register Automata Runner, you can get access to web-based product support, updates, and instructional tips at www.automatarunner.com. You may register Automata Runner at any time from the Windows Start menu and from the program's Help menu.

2.2 Troubleshooting the Installation

Tips on troubleshooting the installation are given in Table 1.

Table 1: Troubleshooting

If you see this symptom...	**...try this.**
The installation fails partway through with an error such as "disk space exhausted."	Check that your computer has 100MB of free disk space, and if it does not, delete and compress files until you have enough space.
The installation fails with a Windows crash message, such as "access denied."	Verify that you have uninstalled any previous version of the software and that you have closed all applications other than the installation program. Also check that the installation directory is not write-protected.
Automata Runner fails with a Windows crash message indicating that virtual memory is exhausted.	Verify that you have 1GB of physical memory and the amount of virtual memory recommended by Windows, and close all applications other than Automata Runner.

If these solutions do not solve the problem, contact product support at www.automatarunner.com.

2.3 Uninstalling the Program

You can uninstall the Automata Runner program from the Windows Start button by clicking Control Panel, then Add or Remove Programs, then Automata Runner, then Remove.

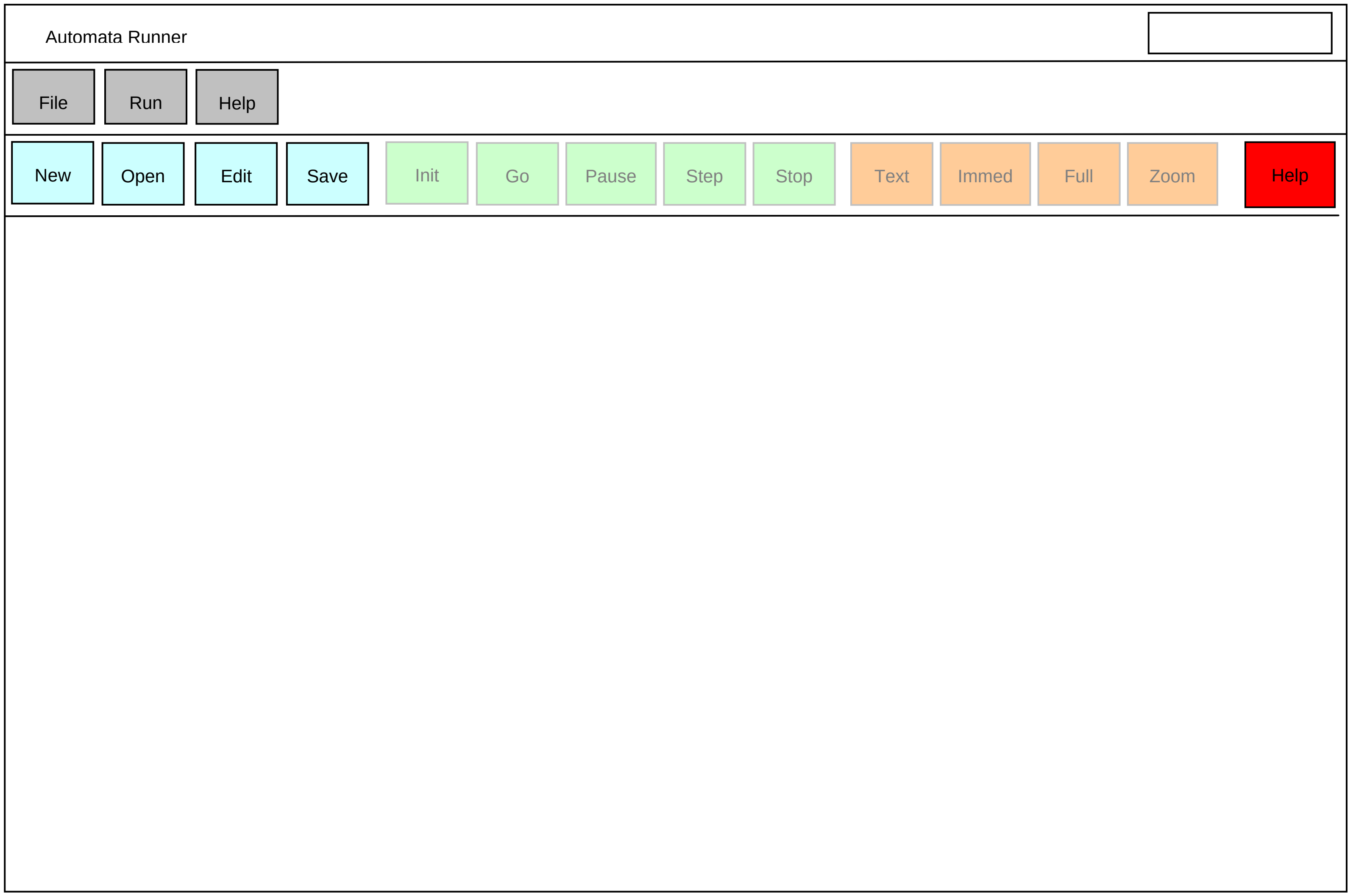
2.4 Running the Program

You can access the program from the Windows Start button by clicking All Programs, then Automata Runner, then Run Automata Runner. This menu item brings up the initial screen of the Automata Runner program, as shown in Figure 1. For a summary of the program buttons and menu items, see the section Command Summary.

From the Windows Start Menu, you can access a number of additional capabilities, all of which are also available from the program Help menu.

- Register Automata Runner. This menu item registers the program with the manufacturer and makes available online help, updates, and instructional tips.
- Update Automata Runner. This menu item downloads updates and bug fixes from the manufacturer's web site.
- Open User's Guide. This menu item brings up the *User's Guide.*
- Open Study Guide. This menu item brings up the *Study Guide.* If this is the Instructor's version of Automata Runner, the program will request the instructor's name and password as entered during installation. If you enter these, the instructor-specific *Study Guide* content will be available; otherwise, it will be blocked.

Figure 1: Main Screen



3 Attributes of an Automaton

A mathematical abstraction such as an automaton consists of a set of attributes and a set of rules by which they are interpreted. The key attributes of an automaton are:

- Automaton Class
- State Set
- Symbol Set
- Memory Model
- Start Configuration
- Next State Function
- End Configuration

The basic operation of an automaton is to (1) detect a change in input; (2) calculate what state to enter as a result of that input; (3) change state; and optionally (4) emit an output in response to the input and to the change in state.

3.1 Automaton Class

In Automata Runner, the class is one of the following:

- Finite State Machine (FSM)
- Stack Machine (SM)
- Turing Machine (TM)
- Cellular Automaton (CA)
- Von Neumann Machine (VNM)

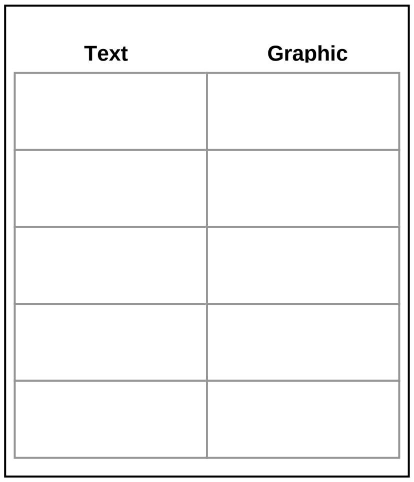
The class of an automaton defines how the other attributes are interpreted and what the automaton's memory model is.

3.2 State and Symbol Sets

In Automata Runner, the state and symbol sets can consist of any text or graphic element. You enter pairs of symbols, text and graphic, into a scrollable table of data entry cells displayed in rows by the program, as shown in Figure 2.

The table will automatically add a row when you have entered data in every previously existing row. If you double-click on a table, the program will bring it up in a separate window.

Figure 2: State/Symbol Data Entry Table



The following rules hold:

- By default, a single row stands for a single state or symbol element, even if it consists of more than one character.
- For every text element, there must be a corresponding graphic element, and for every graphic element, there must be a text element. If you enter an element of one type in a new row, the program will create an element of the other type in the other cell in the row. After you have entered the initial elements, you can edit the text cells and graphic cells individually.
- All cells, whether text or graphic, must be distinct from the others in the same column. A non-unique cell produces an error, indicated by the string !ERR! in the cell. You must eliminate the error before you can run the automaton.

3.2.1 Text Cells

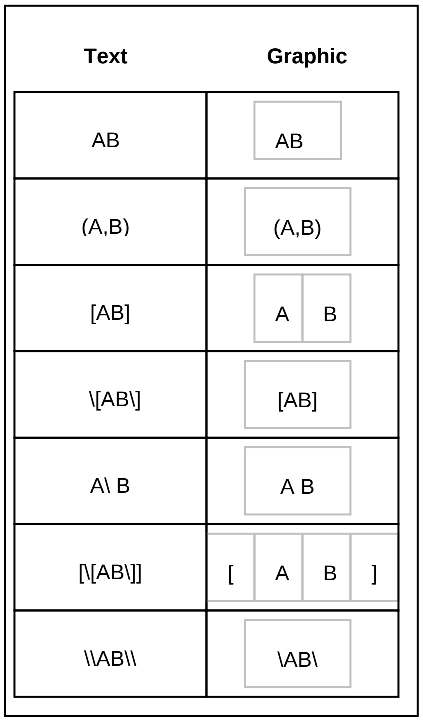
You enter text by clicking on a text cell and entering it as with any Windows program. If the text is longer than the initial width of the cell, the cell will scroll horizontally. When you are entering data into a text cell, you should be aware of the following rules:

- If you type or paste a text element into the text cell of a new row, by default the program will enter a graphic identical to that element into the corresponding graphic cell.
- If a text cell contains the special characters blanks (' ') and tabs ('	'), by default the program removes them from the text and from the corresponding graphic cell when you are finished with the text cell.
- If a text cell contains a string of characters surrounded by the special characters brackets ('[',']'), the program interprets each character within the brackets as a single element, and creates a series of corresponding graphics within the graphic cell. If only one of a pair of brackets appears in the string, there is an error, indicated by the string !ERR! in both the text and graphic cells. You must eliminate the error before you can run the automaton.

- If a text cell contains the special character backslash ('\') before any character (including the special characters), the program reads the character only as that character and not as a special character. This is the only way you can include special characters such as blanks, tabs, brackets, and backslashes in a cell.

A symbol table created by text entry is shown in Figure 3.

Figure 3: Text Entry State/Symbol Table



Text	Graphic
AB	AB
(A,B)	(A,B)
[AB]	A | B
\[AB\]	[AB]
A\ B	A B
[\[AB\]]	[| A | B |]
\\AB\\	\AB\

3.2.2 Graphic Cells

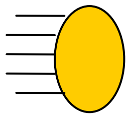
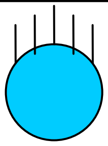
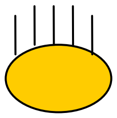
You enter graphics by clicking on a graphics cell. The program will then expand the cell into a separate window and open a graphics editor on its contents. You can draw in the expanded cell or paste graphics into it via the Windows clipboard. Once you have entered graphics in a cell, you can edit them, reposition them, and group and ungroup them as supported by most graphics editors. Note that when you run the automaton, the program will scale the graphics to fit the space allowed for them on the screen, so you should usually enter simple, scaleable graphics.

You enter arbitrary text by typing it into a graphic cell, much as when you enter text into a text cell. However, the program will enter the text as a graphic, not as a string. In particular, the program will give no special interpretation to the special characters (blank, tab, brackets, and backslash).

If you draw or paste a graphic element into the graphic cell of a new row, the program will by default generate a unique arbitrary string in the corresponding text cell. You can then edit this string to some other value.

A symbol table created by graphic entry is shown in Figure 4.

Figure 4: Graphic Entry State/Symbol Table

Text	Graphic
Customer Deposits 25¢	
Machine Outputs: "Please Deposit 25¢"	Please Deposit 25¢
Machine Vends Gumball	
text1	

3.3 *Memory Model*

The memory model of an automaton is the storage assumed by the model.

3.3.1 Finite State Machine

The storage assumed by a finite state machine is limited to the state set. A finite state machine knows the state it has entered, but not necessarily the sequence of state transitions by which it entered that state.

3.3.2 Stack Machine

To be supplied.

3.3.3 Turing Machine

To be supplied.

3.3.4 Cellular Automaton

To be supplied.

3.3.5 Von Neumann Machine

To be supplied.

3.4 Start Configuration

The start configuration is the configuration the automaton is in before its first state transition.

3.4.1 Finite State Machine

The start configuration of a finite state machine consists of the automaton with the current state set to a special state, the Start state, and a sequence of zero or more symbols from the input symbol set in the input buffer, if any. If the automaton generates output, the initial output sequence will be empty.

3.4.2 Stack Machine

To be supplied.

3.4.3 Turing Machine

To be supplied.

3.4.4 Cellular Automaton

To be supplied.

3.4.5 Von Neumann Machine

To be supplied.

3.5 Next-State Function

The next-state function is a table that defines what the automaton will do for each possible state and for each possible input or memory value.

3.5.1 Finite State Machine

For a finite state machine, the next-state function takes as parameters the current state and the current input symbol. Its output is the next state of the FSM and an output symbol (if any).

Automata Runner supports the following kinds of next-state functions, which define subclasses of finite automata:

3.5.1.1 Deterministic

In a deterministic finite automaton, the current state and input symbol yield a unique next state and output symbol, and each state transition must consume an input symbol. This form of finite automaton is the easiest of the three to implement using physical hardware.

3.5.1.2 Nondeterministic

In a nondeterministic finite automaton, the current state and input symbol may yield more than one next state and output symbol. A state transition need not consume an input symbol. It has been proven that for any nondeterministic finite state machine, there is a deterministic FSM that can process the same string of input symbols.

3.5.1.3 Probabilistic

In a probabilistic finite automaton, the current state and input symbol may yield any of a set of next states and output symbols, each with a defined probability. A state transition need not consume an input symbol. The class of probabilistic finite automata is equivalent to nondeterministic finite automata, in the sense that for each probabilistic finite automaton, there is a unique nondeterministic finite automaton.

3.5.2 Stack Machine

To be supplied.

3.5.3 Turing Machine

To be supplied.

3.5.4 Cellular Automaton

To be supplied.

3.5.5 Von Neumann Machine

To be supplied.

3.6 End Configuration

The end configuration is the configuration the automaton has when it completes its operations.

3.6.1 Finite State Machine

A finite state machine reaches its end configuration when the string of input symbols is exhausted. If at that point the machine is in the designated end state (also called the accept state), we say that it has accepted the input string. If it is in some other state, we say that it has rejected the input string.

3.6.2 Stack Machine

To be supplied.

3.6.3 Turing Machine

To be supplied.

3.6.4 Cellular Automaton

To be supplied.

3.6.5 Von Neumann Machine

To be supplied.

4 Defining an Automaton

You define, edit, and save an automaton from the File menu, from the toolbar, or using the hotkeys.

4.1 Creating a New Automaton

To create a new automaton, click File, then New; or click on the New toolbar item; or hit the hotkey **^f^n**. You will see submenus for creating each class of automata currently supported by Automata Runner.

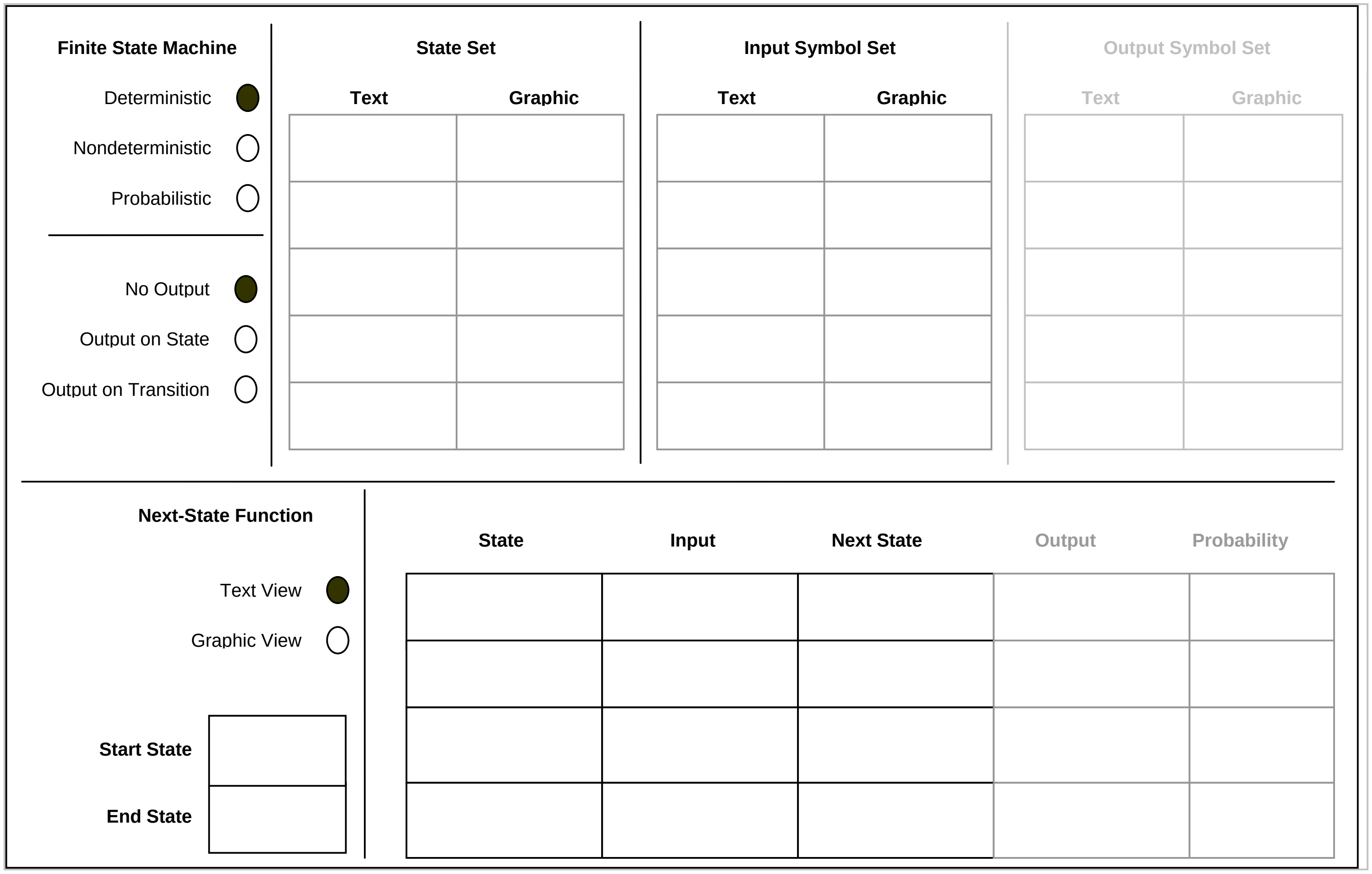
4.1.1 Finite State Machine

A finite state machine consists of the following:

- State set.
- Input symbol set.
- Output symbol set (optional).
- Two special states (the start state and the end state).
- Next-state function (deterministic, nondeterministic, or probabilistic; with or without output).

When you select Finite State Machine from the New menu, the program loads the data entry table shown in Figure 5 into the automaton display. If there is no output symbol set or state-transition probability, the program will disable the relevant columns. When you double-click on a state table, symbol table, or next-state function table, the program will bring the table up in a separate window.

Figure 5: Finite State Machine Data entry table



Finite State Machine

[✓ Deterministic] ●
Nondeterministic ○
Probabilistic ○

[✓ No Output] ●
Output on State ○
Output on Transition ○

State Set

Text	Graphic

Input Symbol Set

Text	Graphic

Output Symbol Set

Text	Graphic

Next-State Function

[✓ Text View] ●
Graphic View ○

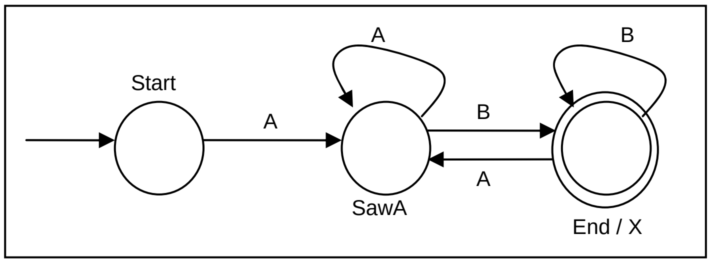
Start State	
End State	

State	Input	Next State	Output	Probability

To enter the data for a finite state machine, you follow the following steps:

1. Select whether you are defining a deterministic, nondeterministic, or probabilistic finite state machine.
2. Select whether you expect to produce output.
3. Enter states and symbols in the symbol tables as described in the section State and Symbol Sets.
4. Select your view of the next-state function. By default, you will see the text view, as shown in Figure 5. In graphic mode, the next-state table will be replaced by a scrollable window containing the automaton in the same graphic arrangement you will see at runtime in Graphic mode. Note that it is usually easier to enter the next-state function in text view than in graphic view. However, you should switch to graphic view at some point and arrange the circles and arrows to your liking. If you do not do so, Automata Runner will try to arrange the circles and arrows for you, but the results may not be what you expect.
5. Enter the next-state function.
   - 5.1. Start and End State Entry. You enter the Start and End states by typing the text entry or by selecting and dragging state table rows into the Start and End cells. In graphic view, the Start state is marked by an arrow passing into it from nowhere. The End state is indicated by a doubled circle.
   - 5.2. Text View State and Symbol Entry. You enter states and symbols into the table by typing the text representing states and symbols into the table cell, or by selecting the table row and pasting the row into the table cell. If there is a probability associated with a state transition, you enter it as a number between 0.0 and 1.0 in the Probability column, or as a percentage.
   - 5.3. Graphic View State Entry. You enter states by selecting the table row or graphic representing a state and pasting it into the display window. A state is represented by a circle containing the state's graphic and labeled by its text name.
   - 5.4. Graphic View State Output Entry. You enter outputs, if any, associated with the state itself, by right-clicking on the state circle, selecting "Output", and dragging or typing one or more output symbols into the box that appears. The output symbols, if any, will be displayed to the right of the state graphic, separated from it by a slash ('/'). See Figure 6, which shows the next-state function for a deterministic FSM that, given a string containing A's followed by a mixture of A's and B's, will output one X for every B read, and accept the string if the last character is B.

Figure 6: Next-State Function for Deterministic FSM with State Output



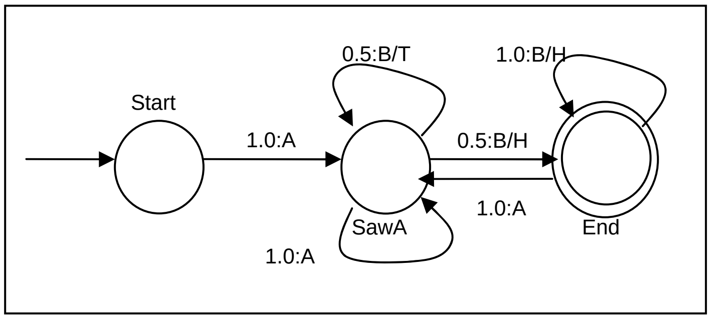
5.5. Graphic View Transition Entry. You create a state transition by right-clicking on one circle, selecting "Transition" in the popup menu, and dragging to another circle. The program will display an arrow connecting the two circles.

5.6. Graphic View Transition Input Entry. You enter an input symbol into a state transition by right-clicking on the arrow, selecting "Input", and dragging or typing one or more input symbols into the box that appears. The input symbol associated with a transition, if any, appears near the arrow.

5.7. Graphic View Transition Output Entry. You enter an output symbol into a state transition by right-clicking on the arrow, selecting "Output", and dragging or typing one or more output symbols into the box that appears. The output symbols, if any, appear to the right of the input symbol, separated from it by a slash ('/').

5.8. Graphic View Probability Entry. You enter the probability associated with a state transition, if any, by entering it as a number between 0.0 and 1.0 in the Probability column, or by entering it as a percentage. It will appear as a number to the left of the input symbol, separated from it by a colon (':'). See Figure 7, which shows the next-state function for a probabilistic FSM that, given a string containing A's followed by a mixture of A's and B's, has a 50-50 chance of outputting a H and going to a state where the string is accepted with the last character a B.

Figure 7: Next-State Function for Probabilistic FSM with Transition Output



4.1.2 Stack Machine

To be supplied.

4.1.3 Turing Machine

To be supplied.

4.1.4 Cellular Automaton

To be supplied.

4.1.5 Von Neumann Machine

To be supplied.

4.2 Opening an Existing Automaton

To open an existing automaton for editing or running, click File, then Open; or click the Open toolbar button; or hit the hotkey **`^f^o`**. This will bring up a standard Windows file manager dialog box. Browse to the directory and file you want to open. You can open files under the following file formats, provided they contain information in the format Automata Runner expects.

- Automata Runner format, file extension: ".atr". This is the default format for opening and saving an automaton.
- XML format, file extension: ".xml". These are XML files saved with a set of tags generated by Automata Runner. If you select an XML file that contains XML in an unexpected format, Automata Runner will raise an error.
- Delimited text format, file extensions: ".txt" and ".csv". These are delimited text files saved in a format generated by Automata Runner. If you select a delimited text file that contains text in an unexpected format, Automata Runner will raise an error.

Open will bring up the data entry table for the given automaton with all of the fields locked against modification.

4.3 Editing the Current Automaton

To edit the current automaton, click File, then Edit; or click the Edit toolbar button; or hit the hotkey **`^f^e`**. This will bring up the data entry table for the current automaton with all of the fields unlocked.

4.4 Saving the Current Automaton

To save an automaton you have created using Automaton Runner, click File, then Save; or click the Save toolbar button; or hit the hotkey **`^f^s`**. The first time you save, the program will prompt you for a file name and format to save the automaton under. Thereafter, unless you click Save-As, you will save the automaton under the same name and format you opened it under.

To save it under a new name or in a new format, click File, then Save-As; or hit the hotkey **`^f^a`**. The program will prompt you for a file name and format to save the automaton under.

5 Running an Automaton

After you have defined an automaton, you will wish to run it and, if necessary, debug it. Automaton Runner supports the following runtime actions.

- Initializing the Automaton
- Starting the Automaton
- Pausing the Automaton
- Stepping the Automaton
- Stopping the Automaton

You can also set runtime display options, as described in the section Setting Runtime Options.

5.1 Initializing the Automaton

To initialize the current automaton, click Run, then Initialize; or click the Init toolbar button; or hit hotkey **`^r^i`**. All bring up the automaton display. In most cases, this consists of an input entry box, an output display box, and additional graphics as required by the class of automaton and the display mode.

5.1.1 Finite State Machine

The automaton display for a finite state machine consists of an input entry box, an output display box, and a representation of the next-state function of the automaton.

5.1.1.1 Text Mode

If the display mode is Text Mode, by default the automaton display shows the next-state table of the FSM, with the row or rows representing the current state highlighted. No graphic symbols will be visible.

5.1.1.2 Graphic Mode

If the display mode is Graphic Mode, the automaton display shows a state as the state icon overlaying a circle and an input symbol next to an arrow from one state to another. If the FSM associates output with the state itself, the output symbol will be shown to the right of the state icon, separated from it by a slash ('/'). If the FSM associates output with the state transition, the output symbol will be shown to the right of the input symbol (if any), separated from it by a slash ('/'). If there is a probability associated with a state transition, the probability is shown to the left of the input symbol, separated from it by a colon (':').

5.1.2 Stack Machine

To be supplied.

5.1.3 Turing Machine

To be supplied.

5.1.4 Cellular Automaton

To be supplied.

5.1.5 Von Neumann Machine

To be supplied.

5.2 Starting the Automaton

To Start the automaton, click Run, then Go; or click the Go toolbar button; or hit hotkey **`^r^g`**. All three run the automaton against the input in a manner that depends upon the input mode.

In Buffered input mode, you type or drag input symbols into the input entry box, then click Go to run the automaton against those inputs.

In Immediate input mode, you click Go, then type or drag input symbols into the input entry box. The automaton runs as far as it can given that input, then pauses and waits for more input.

5.3 Pausing the Automaton

To Pause the automaton, click Run, then Pause; or click the Pause toolbar button; or hit hotkey **`^r^p`**. All three cause the program to highlight the current symbol in the input entry box, highlight the current state in the display, and pause. This command is enabled only in Buffered input mode.

5.4 Stepping the Automaton

To Step the automaton, click Run, then Step; or click the Step toolbar button; or hit hotkey **`^r^s`**. All three cause the program to highlight the next symbol in the input entry box, process it, transition to the next state as appropriate, and pause. This command is enabled only in Buffered input mode.

5.5 Stopping the Automaton

To Stop the automaton, click Run, then Stop; or click the Stop toolbar button; or hit hotkey **`^r^x`**. All three cause the automaton to stop executing, whether the mode is Buffered or Immediate, and return the automaton to its state as of the Initialize command.

6 Setting Runtime Options

You set the runtime options from the Options menu.

6.1 Setting Display Mode

There are two display modes, Text and Graphic. By default, the mode is Graphic and the display mode Text is shown in the Options menu and in the toolbar. Clicking on the display mode in the Options menu or toolbar, or hitting hotkey **`^o^d,`** will change the mode to the option indicated by the button and change the menu item and the button to that for the other option.

If the display mode is Graphic, the program will display the input symbol graphics in a table, display an animated graphic of the automaton, and indicate the current state of the machine, its memory, and its input and output streams by graphic means.

If the display mode is Text, the program will display the next-state table and highlight the state transition being taken. You will type or drag the input symbols into the input box. Outputs will be shown as text, as will the contents of memory, if any.

6.2 Setting Input Mode

There are two input modes, Immediate and Buffered. By default, the mode is Buffered, and the Immediate button is displayed. Clicking on the button or menu item will change the mode to the option indicated and change the button and menu item to that for the other option.

In Buffered input mode, you type or drag input symbols into the input entry box, then click Go. In Immediate input mode, you click Go, then type or drag input symbols into the input entry box. The automaton runs as far as it can given that input, then pauses and waits for more input.

6.3 Setting Transition Speed

The transition speed is the rate at which the program will display state transitions. You select the transition speed by setting a slider somewhere along a range from Animation (about 0.04 second per transition) to Slow (about 2 seconds per transition). By default, the program sets the transition speed to Medium (about 1 second per transition).

6.4 Setting Automaton Time Out

The time out value is the maximum number of state transitions the automaton may perform during a particular run. When the program detects that an automaton run has exceeded this value, it puts up a dialog box telling you that the automaton has timed out and asking if you want to continue, Yes or No. If you click Yes, the program resets its time out counter and continues to execute. If you click No, the program stops in its current state.

6.5 Setting Screen Mode

There are two screen modes, Full and Partial. The default is Partial, in which the Automata Runner program displays in a window. Clicking on the menu item or button will change the screen mode to the option indicated and change the button and menu item to that for the other option.

If you click on Full, the automaton display area fills the entire screen. To use the controls, you mouse to the top of the screen. The menu and tool bar become visible and you can change the options, click Step, click Go, and so on.

6.6 Setting Display Zoom

The Zoom option is enabled only in Graphic display mode. You use the Zoom option to magnify the area around the current state of the machine by setting a slider somewhere along a range from All States, to One Plus Transitions, to One State Only. Note that if you have set the display to Full Screen, setting zoom to One State will cause the current state graphic to fill the entire screen with no circles or arrows. This is the one condition for which it makes sense to create a large, complex graphic.

By default, the program sets the zoom amount to One Plus Connections for a Finite State Machine display and to 5-7 stack, tape, cell, or memory elements for stack machine, Turing machine, cellular automaton, and Von Neumann automaton displays.

7 Getting Help

You access Automata Runner help from the Help menu.

- About Selection. Select an object, then click Help, then About Selection; or click Help; or hit hotkey **^?^a**. This command displays the context-sensitive Help entry for the selected object.
- Register Automata Runner. Click Help, then Register Automata Runner; or hit hotkey **^?^r**. This command registers the program with the manufacturer and makes available online help, updates, and instructional tips.
- Update Automata Runner. Click Help, then Update Automata Runner; or hit hotkey **^?^u**. This command downloads updates and bug fixes from the manufacturer's web site.
- Open User's Guide. Click Help, then Open User's Guide; or hit hotkey **^?^g**. This command brings up the *User's Guide.*
- Open Study Guide. Click Help, then Open Study Guide; or hit hotkey **^?^s**. This command brings up the *Study Guide.* If this is the Instructor's version of Automata Runner, the program will request the instructor's name and password as entered during installation. If the user enters these, the instructor-specific *Study Guide* content will be available; otherwise, it will be blocked.
- About Automata Runner. Click Help, then About Automata Runner; or hit hotkey **^?^b**.This command displays version, build, and copyright data for the Automata Runner program and documentation.

8 Running a Simple Finite State Machine

Suppose you wish to emulate a simple vending machine that dispenses gumballs (25¢) or a toy car (50¢). The data entry table for a simple nondeterministic finite automaton that will model this interaction is shown in Figure 8: Data Entry Table for FSM Modeling

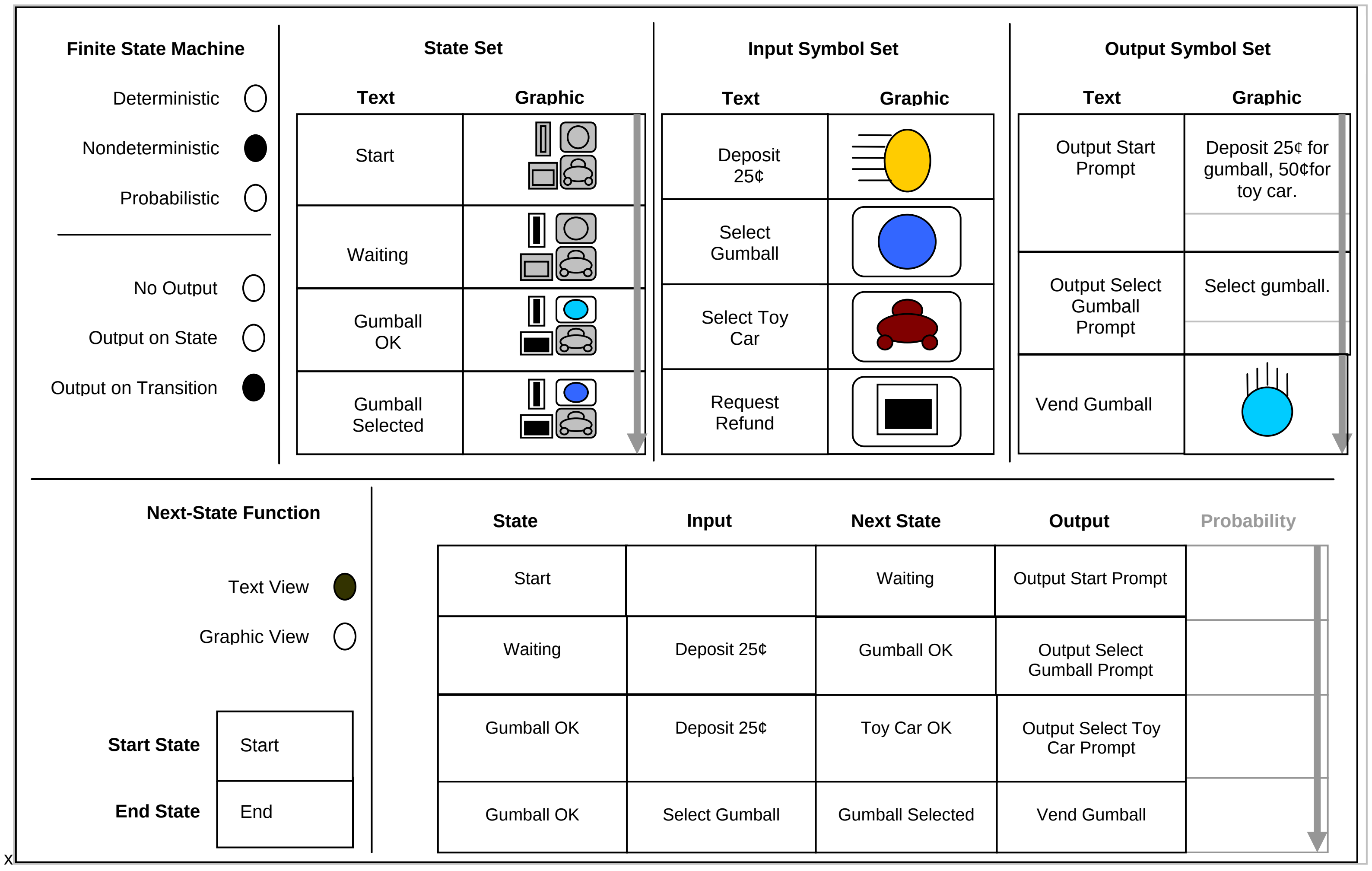
Vending Machine. For simplicity, we will assume that an unlimited supply of gumballs and toy cars are available. (To model a limited supply, we would need a stack machine, a Turing machine, or a Von Neumann machine.)

8.1 Defining the Automaton

In Figure 8, note that due to space constraints, the state, symbol, and next-state tables are not fully visible. Expanded forms of these tables, accessed by double clicking the table, are shown in the following figures:

- Figure 9: State Set for Vending Machine
- Figure 10: Input Symbol Set for Vending Machine.
- Figure 11: Output Symbol Set for Vending Machine.
- Figure 12: Text Next-State Function for Vending Machine.
- Figure 13: Graphic Next-State Function for Vending Machine

Figure 8: Data Entry Table for FSM Modeling Vending Machine



Finite State Machine

- Deterministic ○
- Nondeterministic ●
- Probabilistic ○

- No Output ○
- Output on State ○
- Output on Transition ●

State Set

Text	Graphic
Start	
Waiting	
Gumball OK	
Gumball Selected	

Input Symbol Set

Text	Graphic
Deposit 25¢	
Select Gumball	
Select Toy Car	
Request Refund	

Output Symbol Set

Text	Graphic
Output Start Prompt	Deposit 25¢ for gumball, 50¢for toy car.
Output Select Gumball Prompt	Select gumball.
Vend Gumball	

Next-State Function

- Text View ●
- Graphic View ○

Start State	Start
End State	End

State	Input	Next State	Output	Probability
Start		Waiting	Output Start Prompt	
Waiting	Deposit 25¢	Gumball OK	Output Select Gumball Prompt	
Gumball OK	Deposit 25¢	Toy Car OK	Output Select Toy Car Prompt	
Gumball OK	Select Gumball	Gumball Selected	Vend Gumball	

x

Figure 9: State Set for Vending Machine

State Set

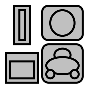
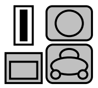
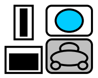
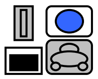
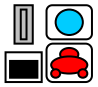
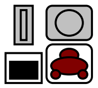
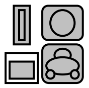
Text	**Graphic**
Start	
Waiting	
Gumball OK	
Gumball Selected	
Toy Car OK	
Toy Car Selected	
End	

Figure 10: Input Symbol Set for Vending Machine

Input Symbol Set

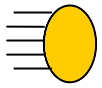
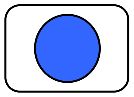
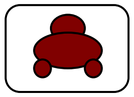
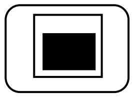
Text	**Graphic**
Deposit 25¢	
Select Gumball	
Select Toy Car	
Request Refund	

Figure 11: Output Symbol Set for Vending Machine

Output Symbol Set

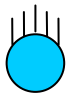
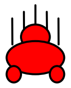
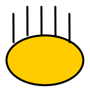
Text	Graphic
Output Start Prompt	25¢ for gumball, 50¢ for toy car.
Output Select Gumball Prompt	Select gumball.
Vend Gumball	
Output Select Toy Car Prompt	Select toy car or gumball.
Vend Toy Car	
Refund 25¢	
Output Goodbye	Thanks, come again!

Figure 12: Text Next-State Function for Vending Machine

State	Input	Next State	Output
Start		Waiting	Output Start Prompt
Waiting	Deposit 25¢	Gumball OK	Output Select Gumball Prompt
Gumball OK	Deposit 25¢	Toy Car OK	Output Select Toy Car Prompt
Gumball OK	Select Gumball	Gumball Selected	Vend Gumball
Gumball OK	Request Refund	End	Refund 25¢
Toy Car OK	Select Toy Car	Toy Car Selected	Vend Toy Car
Toy Car OK	Select Gumball	Gumball Selected	Vend Gumball, Refund 25¢
Toy Car OK	Request Refund	End	Refund 25¢, Refund 25¢
Gumball Selected		End	
Toy Car Selected		End	
End		Start	Output Goodbye

Figure 13: Graphic Next-State Function for Vending Machine

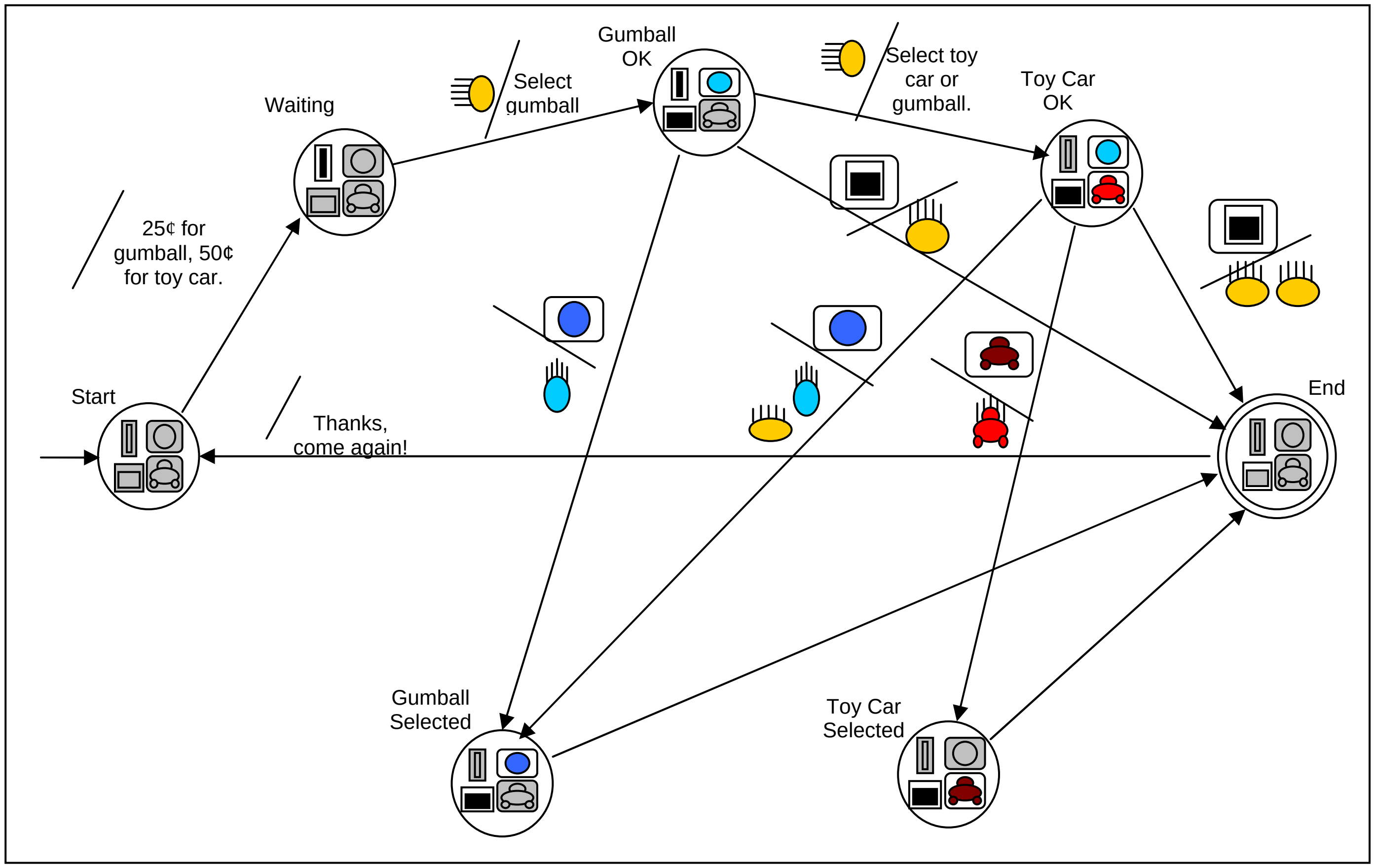
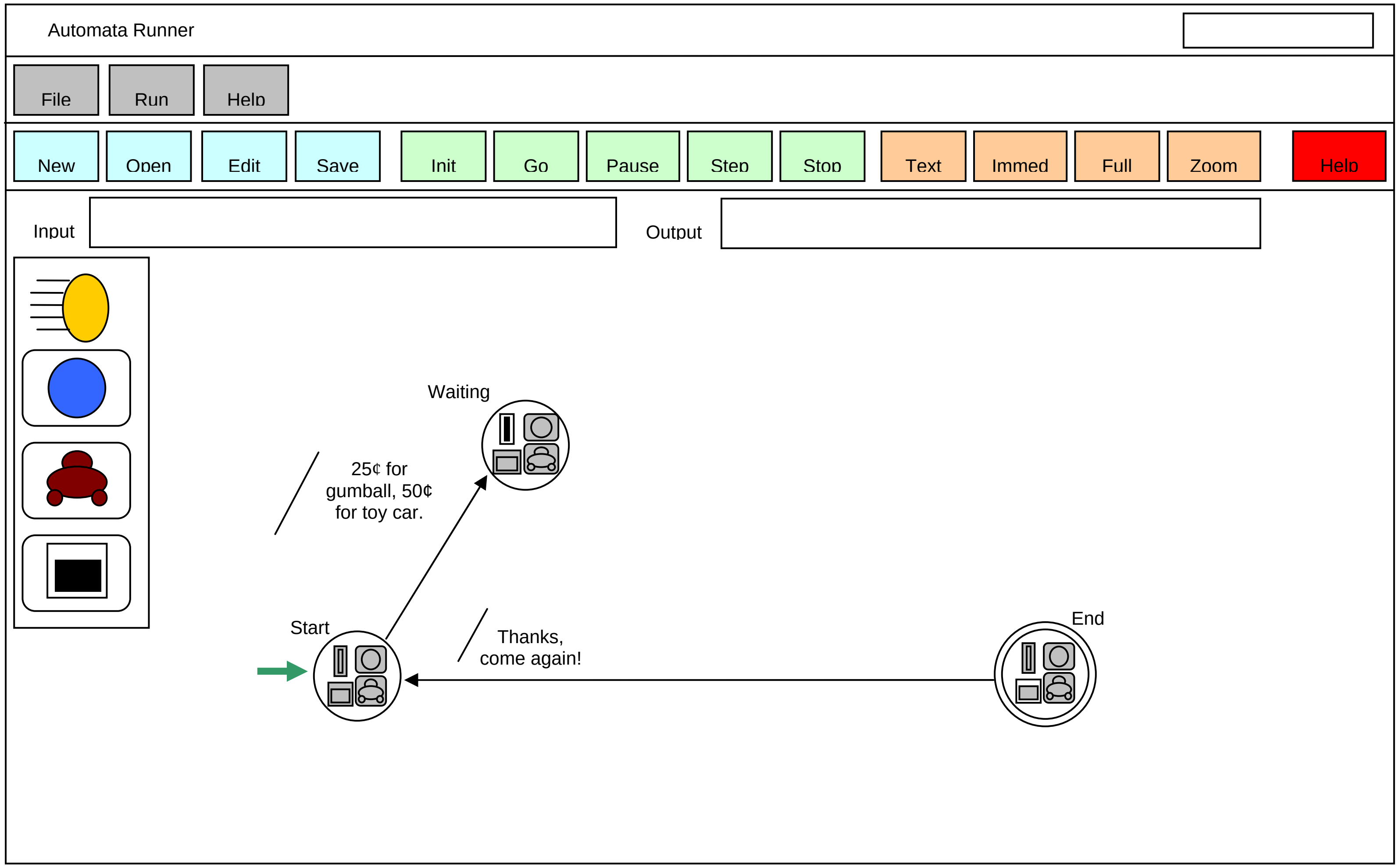


Figure 14: Vending Machine Initial Runtime Screen



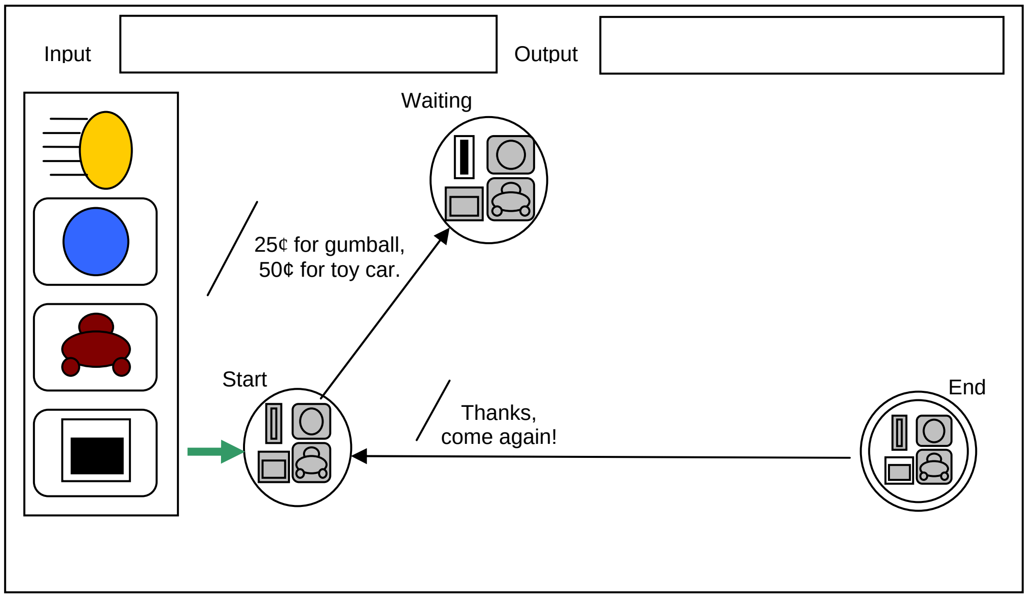
8.2 Running in Graphic Mode

When you bring up the automaton display, you will by default see the runtime screen in Graphic Display Mode at a Zoom level of One Plus Transitions (see Figure 14).

The graphics representing the input states are listed to the left side. The current state and the last state transition arrow are highlighted. In this case, only the arrow leading into the Start state is highlighted, as you have not yet begun to run the automaton and there is no current state.

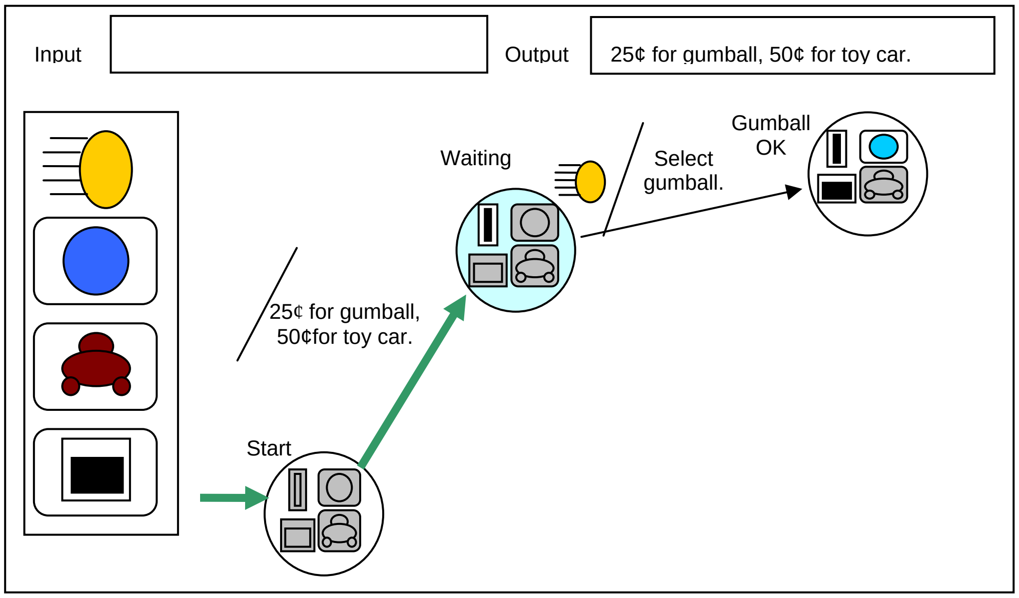
1. Set display mode to Full and input mode to Immediate.

Figure 15: Vending Machine Initial Screen, Zoom Mode



2. Hit Go. You will see first the Start state highlighted, then the nondeterministic state transition to the Waiting state, accompanied by an output (Figure 16).

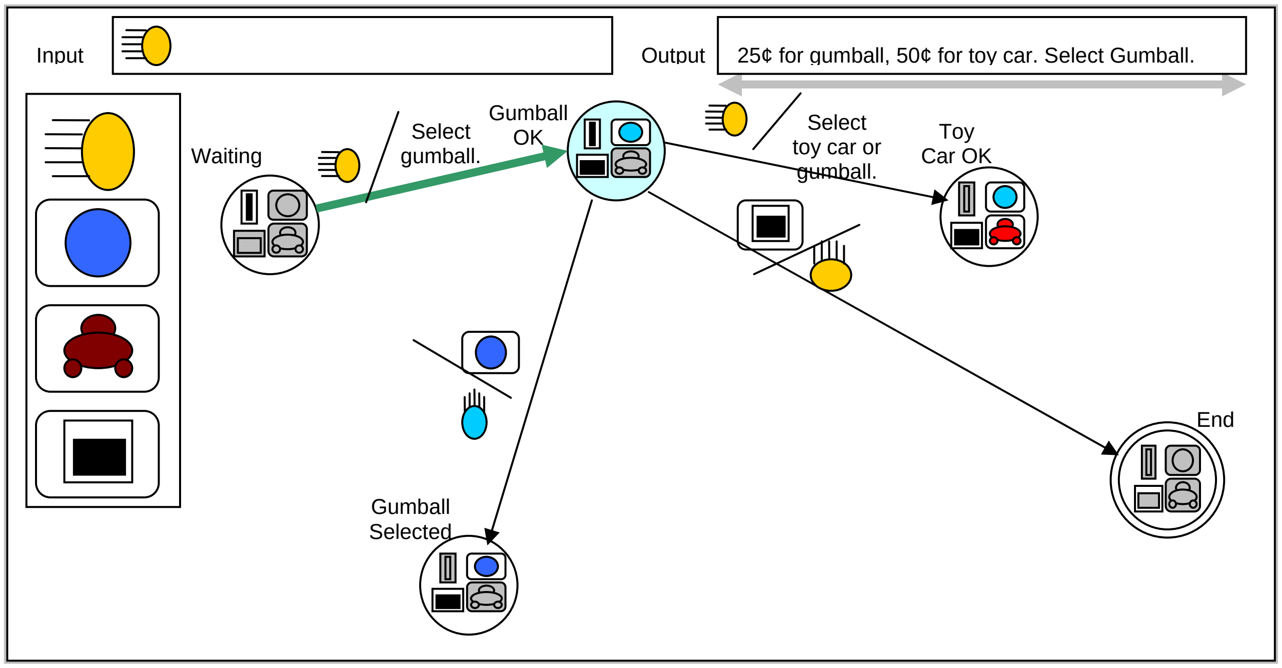
Figure 16: Vending Machine in Wait State



25¢ for gumball, 50¢ for toy car.

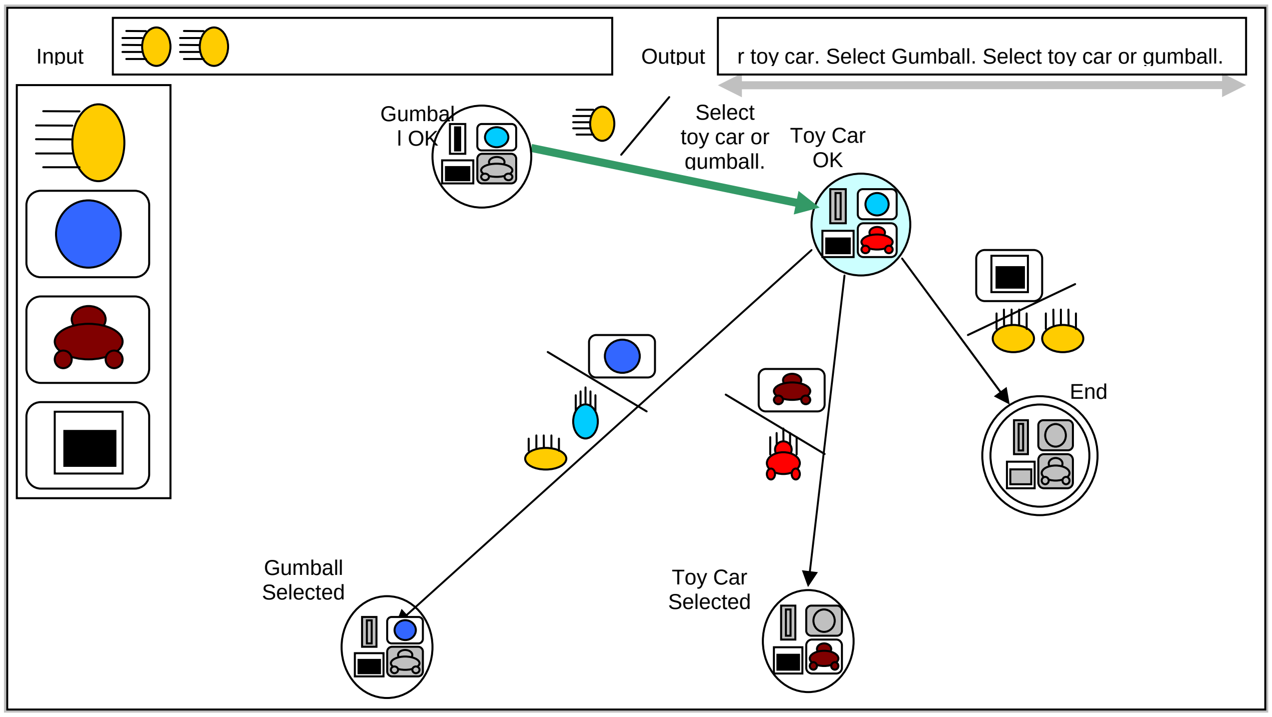
3. You want to buy something, so you deposit 25¢ by clicking on the gold coin icon and go to the Gumball OK state (Figure 17).

Figure 17: Vending Machine in Gumball OK State


Input
Output
25¢ for gumball, 50¢ for toy car. Select Gumball.
Waiting
Select gumball.
Gumball OK
Select toy car or gumball.
Toy Car OK
End
Gumball Selected

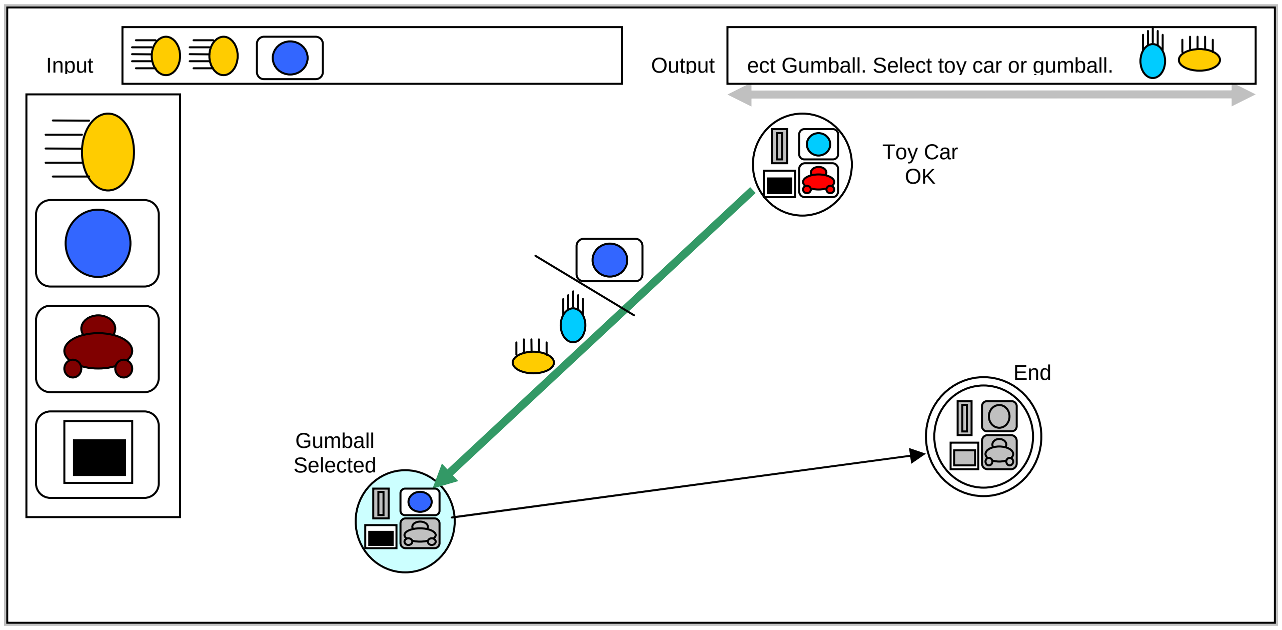
4. You might want a toy car, so you deposit another 25¢ and go to the Toy Car OK state (Figure 18).

Figure 18: Vending Machine in Toy Car OK State.


Input
Output
r toy car. Select Gumball. Select toy car or gumball.
Gumbal l OK
Select toy car or gumball.
Toy Car OK
End
Gumball Selected
Toy Car Selected

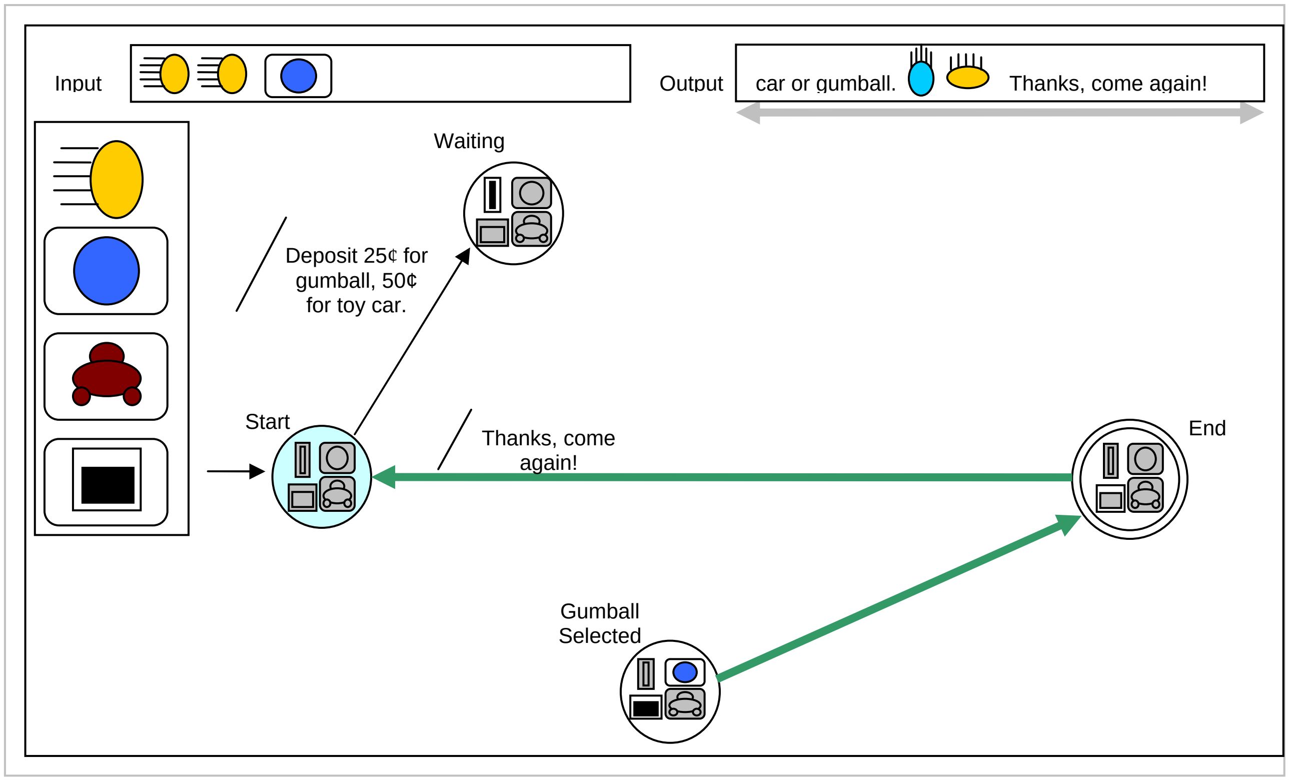
5. You decide you want a gumball after all, so you select the gumball and go to the Gumball Selected state (Figure 19). The output will present a gumball and a 25¢ coin in change.

Figure 19: Vending Machine in Gumball Selected State



6. You then see the nondeterministic transition to the End state, and then, with a message, a nondeterministic transition to the Start state (Figure 20). From there, you go back to Step 2 for another cycle through the automaton.

Figure 20: Vending Machine Returning to Start State

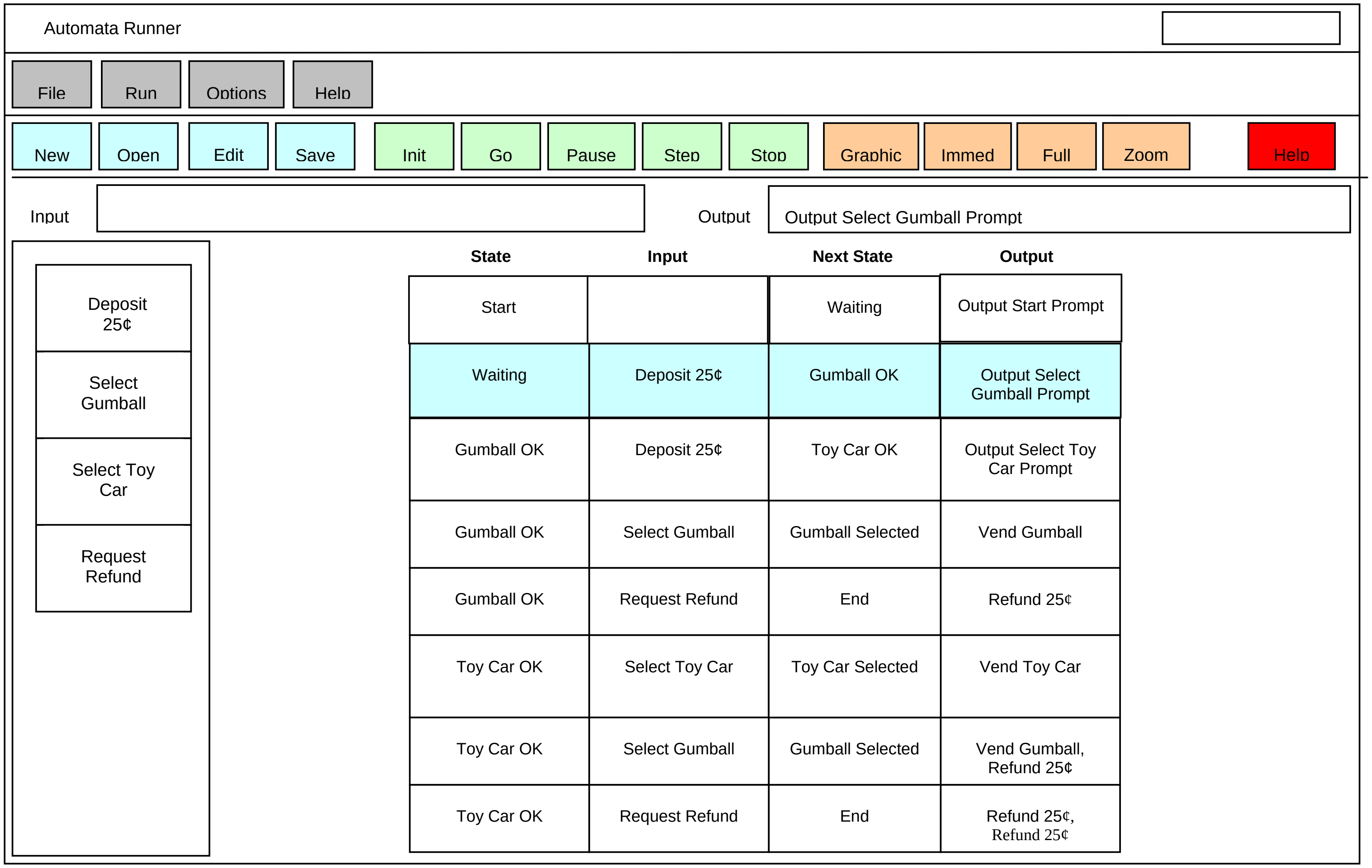


7. You're done with the emulation, so click Stop.

8.3 Running in Text Mode

The Text mode of the emulator is visually simpler. It consists of the text inputs listed down the left side of the screen, and the next-state table of the automaton on the right, with the current state highlighted as shown in Figure 21: Text-Mode Vending Machine in Wait State. You enter input states by either typing one of the input texts, or selecting them and dropping them into the input buffer.

Figure 21: Text-Mode Vending Machine in Wait State.



Output Select Gumball Prompt

State	Input	Next State	Output
Start		Waiting	Output Start Prompt
Waiting	Deposit 25¢	Gumball OK	Output Select Gumball Prompt
Gumball OK	Deposit 25¢	Toy Car OK	Output Select Toy Car Prompt
Gumball OK	Select Gumball	Gumball Selected	Vend Gumball
Gumball OK	Request Refund	End	Refund 25¢
Toy Car OK	Select Toy Car	Toy Car Selected	Vend Toy Car
Toy Car OK	Select Gumball	Gumball Selected	Vend Gumball, Refund 25¢
Toy Car OK	Request Refund	End	Refund 25¢, Refund 25¢

9 Running a Simple Stack Machine

To be supplied.

10 Running a Simple Turing Machine

To be supplied.

11 Running a Simple Cellular Automaton

To be supplied

12 Running a Simple Von Neumann Machine

To be supplied

.

13 Command Summary

The top-level command set of this program is given in Table 2.

Table 2: Command Summary

Menu	**Submenu**	**Toolbar Icon**	**Hot Key**	**Description**
File	New	New	`^f^n`	Creates a new automaton with a user-specified class.
	Open	Open	`^f^o`	Opens a previously saved automaton for editing or running. Initially, the fields are locked against modification.
	Edit	Edit	`^f^e`	Brings up the data entry table for the current automaton with the fields unlocked for modification.
	Save	Save	`^f^s`	Saves the current automaton under the current name and format.
	Save As		`^f^a`	Saves the current automaton under a new user-specified name and format. Not accessible from toolbar.
Run	Initialize	Init	`^r^i`	Brings up the automaton display in its initial state.
	Go	Go	`^r^g`	Runs automata from current state and symbol inputs at Transition Speed.
	Pause	Pause	`^r^p`	Pauses automata run at the current state and next symbol.
	Step	Step	`^r^s`	Transitions to the next state and symbol.

Menu	**Submenu**	**Toolbar Icon**	**Hot Key**	**Description**
	Stop	Stop	`^r^x`	Stops the automaton and returns to its state as of the Initialize command.
Options	Display Mode	Graphic Text	`^o^d`	Switches runtime display mode between Text and Graphic (the default).
	Input Mode	Immed Buffer	`^o^i`	Switches runtime input mode between Immediate and Buffered (the default).
	Transition Speed		`^o^s`	Slider bar used to select the speed at which the emulation changes state. The maximum speed is Animation (0.04 sec/transition); the minimum speed is Slow (2.0 sec/transition); the default is Medium (1.0 sec/transition). Not accessible from toolbar.
	Time Out		`^o^o`	Maximum number of times an automaton may enter a state during a particular run. Not accessible from toolbar.
	Full Screen	Full Partial	`^o^f`	Switches automaton display between Full and Partial. In full mode, controls are hidden at the top of the screen. In Partial mode, the automaton is displayed in a window.
	Zoom	Zoom	`^o^m`	Magnifies area around the current state of the machine along a range from All States to One State.
Help	About Selection	Help	`^?^a`	Displays Help entry for currently selected object.

Menu	Submenu	Toolbar Icon	Hot Key	Description
	Register Automata Runner		^?^r	Registers the program with the manufacturer and makes available online help, updates, and instructional tips. Not accessible from toolbar.
	Update Automata Runner		^?^u	Downloads updates and bug fixes from the manufacturer's web site. Not accessible from toolbar.
	Open User's Guide		^?^g	Opens *User's Guide* as PDF file. Not accessible from toolbar.
	Open Study Guide		^?^s	Opens *Study Guide* as PDF file. If Instructor's Version, requests instructor name and password. Not accessible from toolbar.
	About Automata Runner		^?^b	Displays version, build, and copyright data for Automata Runner program and documentation. Not accessible from toolbar.

14 Index