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

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)

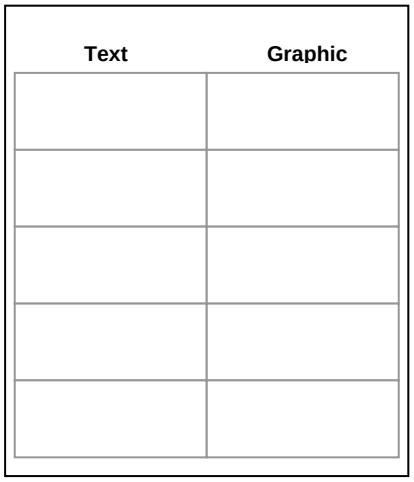
The class of an automaton defines how the other attributes are interpreted and what the automaton's memory model is.

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

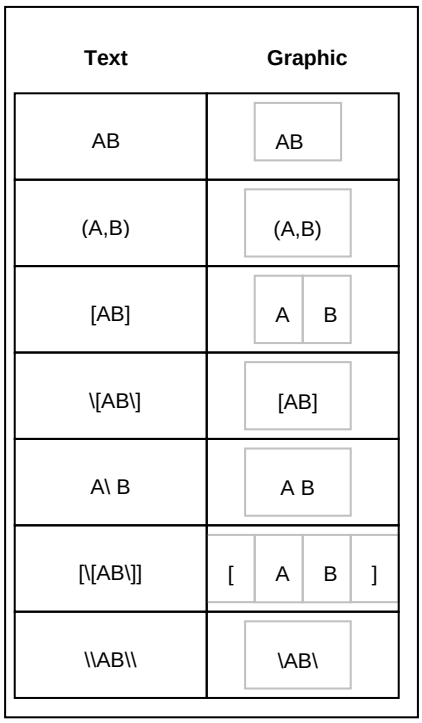
You enter text by clicking on a text cell and entering it as with any Windows program. If the text is longer than the initial width of the cell, the cell will scroll horizontally. When you are entering data into a text cell, you should be aware of the following rules:

- 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



3.2.2 Graphic Cells

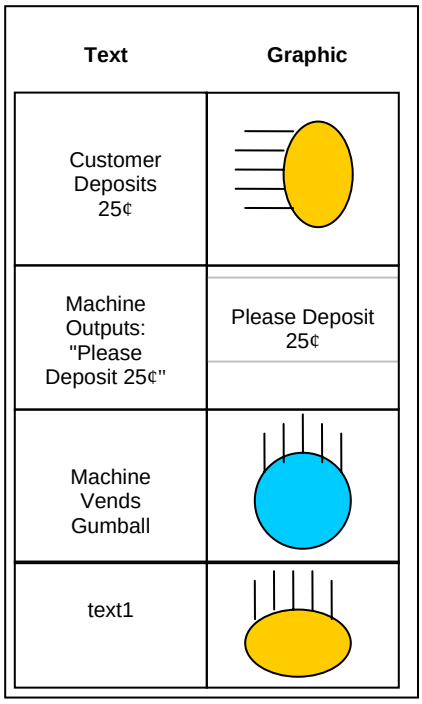
You enter graphics by clicking on a graphics cell. The program will then expand the cell into a separate window and open a graphics editor on its contents. You can draw in the expanded cell or paste graphics into it via the Windows clipboard. Once you have entered graphics in a cell, you can edit them, 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



3.3 Memory Model

3.3.1 Finite State Machine

3.3.2 Stack Machine

3.3.3 Turing Machine

3.3.4 Cellular Automaton

3.3.5 Von Neumann Machine

3.4 Start Configuration

3.4.1 Finite State Machine

3.4.2 Stack Machine

3.4.3 Turing Machine

3.4.4 Cellular Automaton

3.4.5 Von Neumann Machine

3.5 Next-State Function

3.5.1 Finite State Machine

3.5.1.1 Deterministic

3.5.1.2 Nondeterministic

3.5.1.3 Probabilistic

3.5.2 Stack Machine

3.5.3 Turing Machine

3.5.4 Cellular Automaton

3.5.5 Von Neumann Machine

3.6 End Configuration

3.6.1 Finite State Machine

3.6.2 Stack Machine