

Document Plan for *Automata Runner User's Guide*

Version 2.0

David Dahlbacka

Revision History

061114	2.0	First full version.
061114	1.0	Created file.

Contents

Who Will Use <i>Automata Runner User's Guide</i> ?	1
What Is <i>Automata Runner</i> ?	2
Why is a <i>User's Guide</i> Necessary?	2
Outline of <i>Automata Runner User's Guide</i>	3

This is the document plan for the *Automata Runner User's Guide*. It provides a description and user analysis of the program *Automata Runner* and its *User's Guide*, together with an outline of the *User's Guide*.

Who Will Use *Automata Runner User's Guide*?

The primary users of *Automata Runner* and of this guide are undergraduate or graduate students of automata theory. Such students will have a basic knowledge of computer programming and of discrete mathematics. They may be familiar with theorem-proving methods used in automata theory, such as the proof by induction, and with technical applications of automata theory, such as the use of finite-state machines in logic design. They will be familiar with installing and running Microsoft Windows software and with popular-culture applications of automata theory such as those embedded in computer games.

Automata theory may be taught by instructors of mathematics, electrical engineering, or computer science. Although the students are the primary users of the software and *User's Guide*, the course instructors make the decision to adopt the software and the guide. The course instructors will expect the materials to be theoretically accurate and easy to use and will expect them to provide hooks to related technical applications. They may also expect the materials to provide secondary motivational material as well, such as hooks to popular-culture applications.

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 displaying the operation of certain classes of formal automata. Using *Automata Runner*, the user can:

1. Specify or load a formal automaton.
2. Specify a visual display for the automaton's operation.
3. Run and debug the automaton's behavior.
4. Save the automaton for later use.

By providing a visual display of an otherwise abstract process, *Automata Runner* encourages the student to develop intuition about the capabilities, similarities, and differences among automata. Running the automaton and debugging it is very similar to executing a proof by induction, encouraging the student to develop intuition about that method. Finally, displaying the operation of abstract machines may capture the student's personal interest in much the same way as computer games.

Why is a *User's Guide* Necessary?

The primary reason for providing a *User's Guide* is that users expect software to have one, whether provided online or in printed format. The *User's Guide* supports the user in executing the following basic tasks:

- Installing *Automata Runner* in the correct manner on a computer with the correct configuration.
- Specifying an automaton.
- Selecting a visual display for its operation.
- Running the automaton and debugging its behavior.
- Troubleshooting the *Automata Runner* program.

A secondary, but very important, reason for providing a *User's Guide* is to provide supplemental materials related to the educational purpose of the software. The *User's Guide* provides the following:

- A brief overview of the theory of certain classes of automata.
- Guidelines for using the visual display as a starting point for proving theorems about automata.
- Guidelines for customizing the specification and visual display.
- References to publications on automata theory.

Outline of *Automata Runner User's Guide*

The following preliminary outline gives the scope of this document.

1. What is *Automata Runner*?
2. Getting Started
 - 2.1. System Requirements
 - 2.2. Microsoft Windows Installation
 - 2.3. Troubleshooting
3. Specifying an Automaton
 - 3.1. Class
 - 3.1.1. Finite State
 - 3.1.2. Stack
 - 3.1.3. Turing
 - 3.1.4. Cellular
 - 3.2. State Set
 - 3.2.1. Individual
 - 3.2.2. Composite
 - 3.3. Symbol Set
 - 3.3.1. Individual
 - 3.3.2. Composite
 - 3.4. Initial Conditions
 - 3.4.1. Initial State
 - 3.4.2. Initial Symbol Configuration
 - 3.5. Next State Function
 - 3.5.1. Input State and Symbol Configuration
 - 3.5.2. Action
 - 3.5.3. Output State and Symbol Configuration
 - 3.6. Final Conditions
 - 3.6.1. Final State
 - 3.6.2. Final Symbol Configuration
 - 3.7. Options
 - 3.7.1. State Libraries
 - 3.7.2. Symbol Libraries

- 4. Setting the Visual Display
 - 4.1. Mode
 - 4.1.1. Tabular
 - 4.1.2. Graphical
 - 4.2. State
 - 4.2.1. Text
 - 4.2.2. Graphical
 - 4.3. Symbol
 - 4.3.1. Text
 - 4.3.2. Graphical
 - 4.4. Next State Function
 - 4.4.1. Text
 - 4.4.2. Graphical
 - 4.5. Options
 - 4.5.1. Color Palette
- 5. Running and Debugging the Automaton
 - 5.1. Go
 - 5.2. Pause
 - 5.3. Step
 - 5.4. Stop
 - 5.5. Options
 - 5.5.1. Transition Speed
- 6. Storing the Automaton
 - 6.1. New
 - 6.1.1. Automaton
 - 6.1.2. Library
 - 6.2. Open
 - 6.2.1. Automaton
 - 6.2.2. Library
 - 6.3. Save / Save-As
 - 6.3.1. Automaton
 - 6.3.2. Library
 - 6.4. Export

- 6.4.1. Automaton
- 6.4.2. Library
- 7. Appendices
 - 7.1. Automata Theory Overview
 - 7.1.1. Finite State
 - 7.1.2. Stack
 - 7.1.3. Turing
 - 7.1.4. Cellular
 - 7.2. Theorem-Proving Methods
 - 7.2.1. Proof by Induction
 - 7.2.2. Proof by Contradiction
 - 7.2.3. Proof by Reduction
 - 7.3. Applications
 - 7.3.1. Logic Design
 - 7.3.2. Formal Language Parsing
 - 7.3.3. Computer Games
 - 7.4. Customizations
 - 7.4.1. Automaton Representation
 - 7.4.2. Automaton Emulation
 - 7.5. References