

<http://www.CPP.edu/~jrfisher/> → www/prolog_tutorial/contents.html

prolog :- tutorial

© J.R.Fisher

Table of Contents

[Introduction](#)

[1. How to Run Prolog](#)

2. Sample Programs -- [Descriptions](#)

- [2.1](#) Map colorings
 - [2.2](#) Two factorial definitions
 - [2.3](#) Towers of Hanoi puzzle
 - [2.4](#) Loading programs, editing programs
 - [2.5](#) Negation as failure
 - [2.6](#) Tree data and relations
 - [2.7](#) Prolog lists and sequences
 - [2.8](#) Change for a dollar
 - [2.9](#) Map coloring redux
 - [2.10](#) Simple I/O
 - [2.11](#) Chess queens challenge puzzle
 - [2.12](#) Finding all answers
 - [2.13](#) Truth table maker
 - [2.14](#) DFA parser
 - [2.15](#) Graph structures and paths
 - [2.16](#) Search
 - [2.17](#) Animal identification game
 - [2.18](#) Clauses as data
 - [2.19](#) Actions and plans
-

3. How Prolog Works

- [3.1](#) Prolog derivation trees, choices and unification
 - [3.2](#) Cut
 - [3.3](#) Meta-interpreters in Prolog
-

[4. Built-in Goals](#)

- [4.1](#) Utility goals
- [4.2](#) Universals (true and fail)
- [4.3](#) Loading Prolog programs
- [4.4](#) Arithmetic goals
- [4.5](#) Testing types
- [4.6](#) Equality of Prolog terms, unification
- [4.7](#) Control
- [4.8](#) Testing for variables
- [4.9](#) Assert and retract
- [4.10](#) Binding a variable to a numerical value
- [4.11](#) Procedural negation, negation as failure
- [4.12](#) Input/output
- [4.13](#) Prolog terms and clauses as data

- [4.14](#) Prolog operators
 - [4.15](#) Finding all answers
-

5. Search in Prolog

- [5.1](#) The A* algorithm in Prolog
 - [5.2](#) The 8-puzzle
 - [5.3](#) $\alpha\beta$ search in Prolog
-

6. Logic Topics

- [6.1](#) Chapter 6 notes
 - [6.2](#) Positive logic
 - [6.3](#) Convert first-order logic to normal form
 - [6.4](#) A normal rulebase goal interpreter
 - [6.5](#) Evidentiary soundness and completeness
 - [6.6](#) Rule tree visualization using Java
-

7. Introduction to Natural Language Processing

- [7.1](#) Prolog grammar parser generator
 - [7.2](#) Prolog grammar for simple English phrase structures
 - [7.3](#) Idiomatic natural language command and question interfaces
-

8. Prototyping with Prolog

- [8.1](#) Action specification for a simple calculator
 - [8.2](#) Animating the 8-puzzle (§5.2) using character graphics
 - [8.3](#) Animating the blocks mover (§2.19) using character graphics
 - [8.4](#) Java Tic-Tac-Toe GUI plays against Prolog opponent (§5.3)
 - [8.5](#) Structure diagrams and Prolog
-

[References](#)



Use Frames