Chapter 5. Early Heuristic Programs

The Quest for Artificial Intelligence, Nilsson, N. J., 2009.

Lecture Notes on Artificial Intelligence, Spring 2012

Summarized by Ha, Jung-Woo and Lee, Beom-Jin

Biointelligence Laboratory
School of Computer Science and Engineering
Seoul National Univertisy

http://bi.snu.ac.kr

Contents

5.1 The Logic Theorist and Heuristic Search
- The Logic Theorist and Heuristic Search

5.2 Proving Theorems in Geometry
- Proving Theorems in Geometry

5.3 The General Problem Solver
- The General Problem Solver

5.4 Game-Playing Programs
- Game-Playing Programs

Appendix

Overview of Chapter 5

- In early 1960s, many approaches for proving various kinds of logical theorem
  - Symbolic structure
  - Combinatorial explosion $\rightarrow$ heuristic search
  - Geometric problem
    - Subproblems
  - Solving general problems
  - Game-playing program with intelligence

Chapter 5. Early Heuristic Programs

5.1 The Logic Theorist and Heuristic Search

The Logic Theorist and Heuristic Search

- Logic theorist (LT) and symbolic structures
  - LT uses symbolic structures for proving theories.
- Symbolic structure: symbols → lists → lists of lists
  - Simple: symbols and lists (A, 7, Q)
  - More complex: lists of lists of lists… ((B, 3), (A, 7, Q))
  - Transformation:
    (the sum of seven and five) → (7+5) → (12)
  - Transforming structures & searching for problem-solving sequences of transformation: the key of Newell and Simon's idea
  - Example: 8-puzzle game

The Logic Theorist and Heuristic Search

- Combinatorial explosion
  - In case of very large versions of the puzzle
  - Cannot guarantee the generation of the goal state
- Heuristic search
  - A heuristic for the problem: A process that may solve a problem but offers no guarantees of doing so
  - Not all directions but some guided directions
  - No guarantee but elimination of much fruitless efforts
- Emerging of Heuristic programming
- Importance of representation of the problem

Chapter 5. Early Heuristic Programs

5.2 Proving Theorems in Geometry

Proving Theorems in Geometry

- Geometry-theorem-proving
  - Gelernter version (1959)
    - Explicit use of the subgoals (reasoning backward or divide & conquer)
    - A diagram to close off futile search paths
  - Example: equal angle problem → congruent triangle

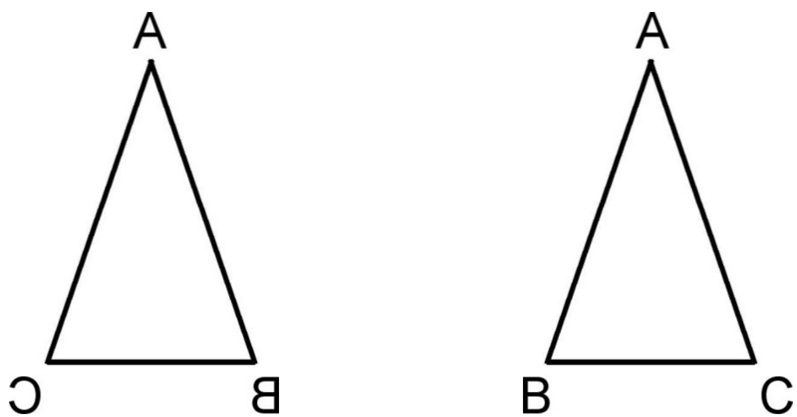

Figure 5.4: A triangle with two equal sides (left) and its flipped-over version (right).

5.3 The General Problem Solver

The General Problem Solver

- General problem solver (GPS)
  - An embodiment of Newell, Shaw, and Simon’s ideas about how humans solve problems
  - Inspired later work in AI and cognitive science
  - An earlier work of LT based on symbolic structures
  - Subproblems strategy like Gelernter’s geometry program
  - Mean-ends analysis
    - GPS compute the difference (subproblems) between the given problem and an already known solved problem
    - Then, it attempts to reduce the difference by some symbol-manipulating “operator” to the initial symbolic structure

Chapter 5. Early Heuristic Programs

5.4 Game-Playing Programs

Game-Playing Programs

- Early works for chess game
  - Shannon's and Newell, Shaw, and Simon's
  - Babbage's book – *The Life of Philosopher*
  - Zues's using Plankalkul
  - Turing: chess-playing game → intelligence of computers
    - Turing vs. chess program http://www.chessgames.com/perl/chessgame?gid=1356927
  - In 1960s, the competent chess program is developed

Game-Playing Programs

- Checkers (draughts)
  - Simple game than chess
  - C. Strachey's check-playing program (1951)
  - A. Samuel's check program (1952)
    - How to get a computer to learn from game experience
    - The first attempt of machine learning

Figure 5.5: Arthur Samuel

- Samuel's check-playing game
  - Similar to the eight-puzzle
  - Use of symbolic tree representing board positions

Game-Playing Programs

- Samuel's check program
  - For each point, all possible configurations are considered
  - 5×10^{20} positions → Infeasible
  - Look only a few moves ahead → Incomplete tree
  - How to choose a move from the incomplete tree?

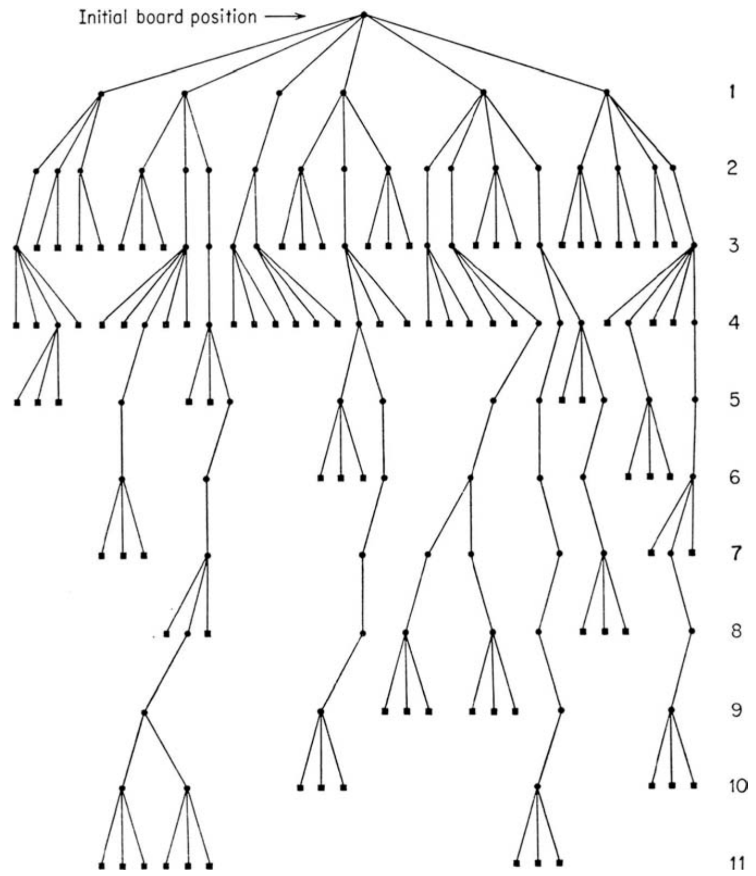

Figure 5.6: An illustrative checkers game tree.

Game-Playing Programs

- Samuel's check program
  - How is the program to choose a move from incomplete tree?
    - Computing a score for the positions at the tips (leaves) of the tree
    - Migrating this score back up to the positions resulting from moves from the current position
  - Measuring the score
    - The points to be awarded to positions at the leaves of the tree based on their overall "goodness"
    - The points contributed by each feature were then multiplied by a "weight"
  - Migration
    - Highest-lowest migration strategy
    - Alpha-beta migration strategy
  - Learning methods
    - Adjustment of the weight values by the scoring system
    - Rote learning

Chapter 5. Early Heuristic Programs

Appendix

The Logic Theorist and Heuristic Search

- Examples of using symbolic structures
  - Problem: start tile state → goal state

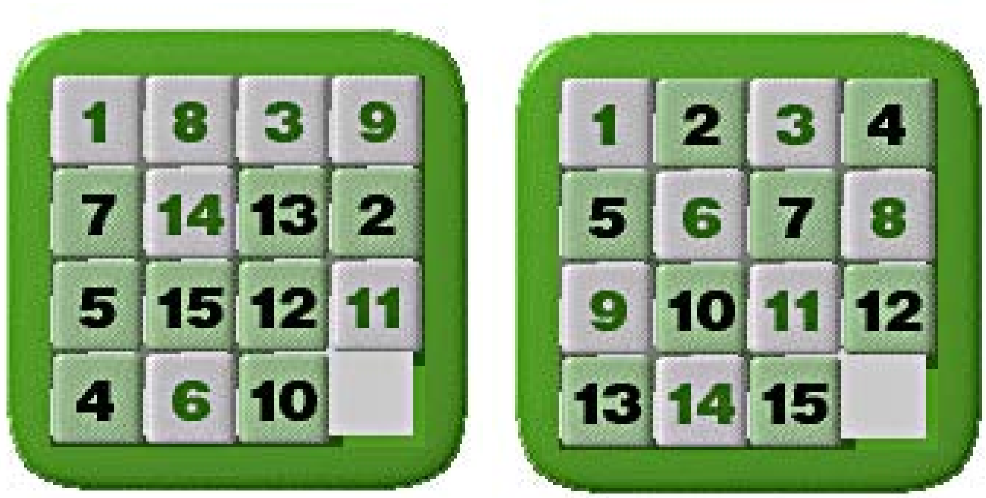

Figure 5.1: Start (left) and goal (right) configurations of a fifteen-puzzle problem

The Logic Theorist and Heuristic Search

- An easier version (3 × 3) of the tile puzzle problem
  - Representation with symbol structures
    - Start: ((2, 8, 3), (1, 6, 4), (7, B, 5))
    - Goal: ((1, 2, 3), (8, B, 4), (7, 6, 5))
  - Make rules for transforming the structures
  - Transform the structures by rules

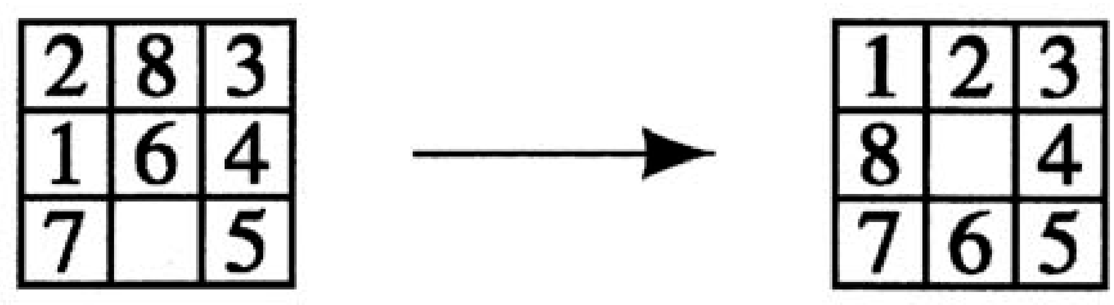

Figure 5.2: The eight-puzzle.

The Logic Theorist and Heuristic Search

- A search tree of symbolic structures by transformation

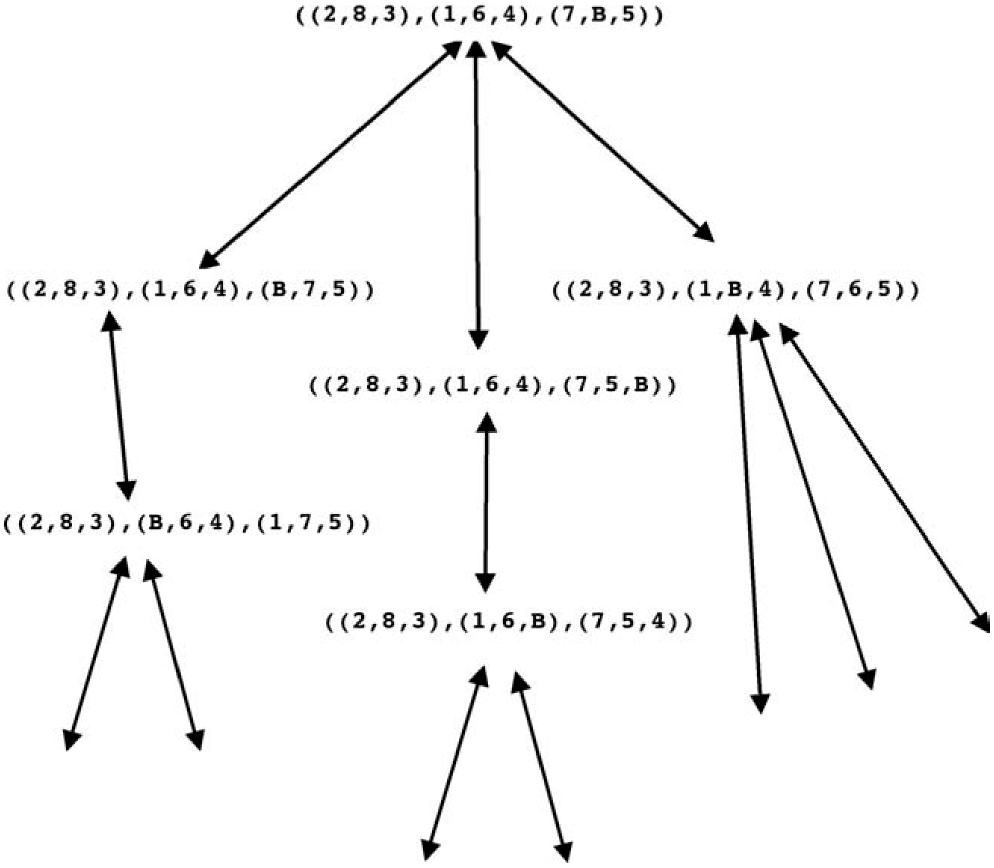

Figure 5.3: A search tree.