

CNG 462 Artificial Intelligence

Assignment 2

Date Handed Out

January 03, 2021

Date Submission Due

Till the end of the day of Jan 15, 2021

Introduction

This assignment aims to help you gain experience with constraint satisfaction problems (CSPs). You will write a program to solve the following problem using backtracking algorithm.

The Crossword Puzzle

A crossword is “a word puzzle and word search game that usually takes the form of a square or a rectangular grid of white- and black-shaded squares. The game's goal is to fill the white squares with letters, forming words or phrases, by solving clues, which lead to the answers. In languages that are written left-to-right, the answer words and phrases are placed in the grid from left to right and from top to bottom. The shaded squares are used to separate the words or phrases” [1].

The crossword puzzle is depicted in Fig. 1. In this example, for simplicity, we assume that candidates of words are specified. Since a five-letter word must be placed in 1-across, possible candidates are {Hoses, Laser, Sails, Sheet, Steer}. If we choose 'Hoses' for 1-across, we can only choose a five-letter word that begins with the letter 's' for 2-down, i.e., 'Sails', 'Sheet', or 'Steer'.

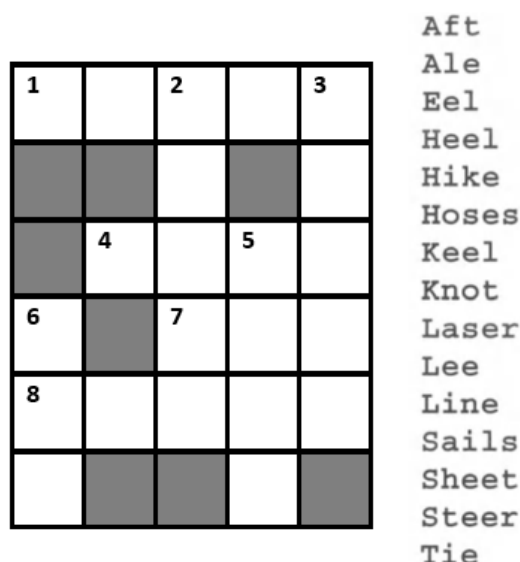


Fig. 1. Crossword Puzzle

This problem can be formalized as a CSP. If all constraints are binary (i.e., between two variables), a CSP can be represented as a network, in which a node represents a variable, and a link between nodes represents a constraint between the corresponding variables.

1. Programming Tasks

1. Implement a backtracking algorithm. A backtracking algorithm is a basic, systematic search algorithm for solving CSPs. In a backtracking algorithm, a value assignment to a subset of variables that satisfies all of the constraints within the subset is constructed. This subset is called a partial solution. Initially, a partial solution is an empty set. A partial solution is expanded by adding new variables one by one, until it becomes a complete solution.
2. Implement a heuristic for your solution.
3. Provide a way to show the assignments step by step. In other words, your implementation must show the steps of the backtracking algorithm.

2. User Interface (UI)

Don't spend too much time on the graphics and UI development. A command line representation is enough so long as it is understandable.

3. Programming Language Specifications

Your program can be written in Java, C, C#, C++, or Python.

4. Submission Requirements

Along your implementation, you also need to submit a short report that includes the following:

- The full CSP problem formulation. A clear description must be given in a formal notation.
- The algorithm and any heuristics that you apply. Explain your algorithm and heuristics.
- The implementation details of your code.
- A brief description (approximately 1 page in length) of your experience programming for and participating in the tournament.

5. Submission

You need to submit a ZIP file (firstname_lastname.zip, e.g., okan_topcu.zip) including the following:

1. Report in PDF format (see Sect. 4 above).
2. Your source code for the game.
3. `Readme.txt` file for your source code. This should include a short description of your program and it should explain how to compile and run your code, please include your name, surname and student id at the top.

To submit your assignment, simply select the appropriate assignment link from the ODTUCLASS page. Upload your zip file and click submit (clicking send is not enough). Please make sure all source files are included in your zip file when submitted. A program that does not compile as submitted will be given 0 points. Only your **final submission** will be graded. Remember there is no late submission for this assignment.

6. Academic Honesty

You may find implementations of a puzzle solver on the Web, please do not download and submit any of those implementations. You can refer to their solution, but you really need to submit your own solution. Please see syllabus for the details of academic honesty.

You will see that it will be so much fun to implement a solution. And this assignment is a chance to test your AI and programming skills. Have fun!

7. Grading

Grading Point	Total (100)
A working demo of your system	/30
The algorithm and any heuristics that you apply. Explain your algorithm and heuristics.	/20
Provide a way to show the assignments step by step. In other words, your implementation must show the steps of the backtracking algorithm	/20
The full CSP problem formulation. A clear description must be given in a formal notation.	/15
The implementation details of your code. Code must be readable and understandable.	/10
A brief description (approximately 1 page in length) of your experience programming for and participating in the tournament.	/5
TOTAL	/100

You are free to work closely with other students, but each student must code up his/her own agents and design his/her function.

References

- [1] Wikipedia, «Crossword,» Wikipedia, [Çevrimiçi]. Available: <https://en.wikipedia.org/wiki/Crossword>. [Erişildi: 3 January 2021].