Lab Assignment 4
Single-Agent Search

T-622-ARTI,Artificial Intelligence, 2023-1

Reykjavik University - School of Computer Science, Menntavegi 1, IS-101 Reykjavík, Iceland

Jonas Hagg jonash22@ru.is
Philipp Zimmermann philipp22@ru.is
Jonathan Lutz jonathan22@ru.is

February 20, 2023

Abstract

This is the report for the fourth lab assignment of the artificial intelligence course.

1 Tasks

1.1 Task 1

We choose to model the CSP with 25 Variables. The variables represents each of the House color, nationalities, pets, drinks or cigarettes. Stated below are all 25 variables we used:

1. Red
2. Green
3. Ivory
4. Yellow
5. Blue
6. Englishman
7. Spaniard
8. Norwegian
9. Ukrainian
10. Japanese
11. Old Gold
12. Kools
13. Chesterfields
14. Lucky Strike
15. Parliaments
16. Water
17. Orange juice
18. Tea
19. Coffee
20. Milk
21. Zebra
22. Dog
23. Fox
24. Snails

25. Horse

The Domains of the Variables are in this case the possible House Numbers from 1 to 5. When Modeling the CSP like this, each variable has the same Domain space, which heavily reduces the complexity of Constraints. Because all Variables have the same domain we can directly compare each variable by their value.
Out of each statement of the puzzle, wecreated following Constraints for the CSP:

- Englishman = Red
- Spaniard = Dog
- Coffee = Green
- Ukrainian = Tea
- Green = Ivory + 1
- Old Gold = Snails
- Kools = Yellow
- Milk = 3
- Norwegian = 1
- | Chesterfields - Fox | = 1
- | Kools - Horse | = 1
- Lucky Strike = Orange juice
- Japanese = Parliaments
- | Norwegian - Blue | - 1

Out of the context of the puzzle we know, that each house needs to have unique properties. Therefore we also added following Constrains:

- AllDifferent(Red, Green, Ivory, Yellow, Blue)
- AllDifferent(Englishman, Spaniard, Norwegian, Ukrainian, Japanese)
- AllDifferent(Old Gold, Kools, Chesterfields, Lucky Strike, Parliaments)
- AllDifferent(Water, Orange juice, Tea, Coffee, Milk)
- AllDifferent(Zebra, Dog, Fox, Snails, Horse)

1.2 Task 2

The state space for a CSP can be defined as the cartesian product of the domains of all variables. Therefore, the state space is:

$$d^n = 5^{25} \approx 2.98 \cdot 10^{17} \tag{1}$$

The upper estimate for the size of the search tree is the same as the state space (5^{25}) because, if all variables are assigned one value at a time, and constraints are checked after each assignment, the size of the search tree also is the cartesian product of the domains of all variables.

The lower estimate for the size of the search tree is 25, if only one solution is applicable.

1.3 Task 3

Solution found by the python implementation:

- Red = 3
- Green = 5
- Ivory = 4
- Yellow = 1
- Blue = 2
- Englishman = 3
- Spaniard = 4
- Norwegian = 1
- Ukrainian = 2
- Japanese = 5
- Old Gold = 3
- Kools = 1
- Chesterfields = 2
- Lucky Strike = 4
- Parliaments = 5
- Water = 1
- Orange juice = 4
- Tea = 2
- Coffee = 5
- Milk = 3
- Zebra = 5
- Dog = 4
- Fox = 1
- Snails = 3
- Horse = 2

1.4 Task 4

Running the Python Implementation took between 0.00 and 0.01 seconds to complete. In this short amount of time it is impossible to search through the entire Search Tree, therefore we assume that only a fraction of the tree has been visited during the search. This is due to good pruning of the Tree Nodes depending on our Constrains.
The Implementation also returned a branching of 0 (*branches: 0*). This validates our assumption that the pruning of nodes is really high. In fact the Search is able to find the single possible Solution without any backtracking.