Assignment 2 AC3 and Backtrack search

1. Your program’s solution for each of the Sudoku boards shown in Figures 1a-1d.

a)

7 8 4 | 9 3 2 | 1 5 6

6 1 9 | 4 8 5 | 3 2 7

2 3 5 | 1 7 6 | 4 8 9

------+-------+------

5 7 8 | 2 6 1 | 9 3 4

3 4 1 | 8 9 7 | 5 6 2

9 2 6 | 5 4 3 | 8 7 1

------+-------+------

4 5 3 | 7 2 9 | 6 1 8

8 6 2 | 3 1 4 | 7 9 5

1 9 7 | 6 5 8 | 2 4 3

b)

8 7 5 | 9 3 6 | 1 4 2

1 6 9 | 7 2 4 | 3 8 5

2 4 3 | 8 5 1 | 6 7 9

------+-------+------

4 5 2 | 6 9 7 | 8 3 1

9 8 6 | 4 1 3 | 2 5 7

7 3 1 | 5 8 2 | 9 6 4

------+-------+------

5 1 7 | 3 6 9 | 4 2 8

6 2 8 | 1 4 5 | 7 9 3

3 9 4 | 2 7 8 | 5 1 6

c)

1 5 2 | 3 4 6 | 8 9 7

4 3 7 | 1 8 9 | 6 5 2

6 8 9 | 5 7 2 | 3 1 4

------+-------+------

8 2 1 | 6 3 7 | 9 4 5

5 4 3 | 8 9 1 | 7 2 6

9 7 6 | 4 2 5 | 1 8 3

------+-------+------

7 9 8 | 2 5 3 | 4 6 1

3 6 5 | 9 1 4 | 2 7 8

2 1 4 | 7 6 8 | 5 3 9

d)

4 3 1 | 8 6 7 | 9 2 5

6 5 2 | 4 9 1 | 3 8 7

8 9 7 | 5 3 2 | 1 6 4

------+-------+------

3 8 4 | 9 7 6 | 5 1 2

5 1 9 | 2 8 4 | 7 3 6

2 7 6 | 3 1 5 | 8 4 9

------+-------+------

9 4 3 | 7 2 8 | 6 5 1

7 6 5 | 1 4 3 | 2 9 8

1 2 8 | 6 5 9 | 4 7 3

1. The domains after running the AC-3 algorithm for each of the four boards.

Easy board: A grid of numbers on a black background

Description automatically generated

Medium board:

A grid of numbers on a black background

Description automatically generated

Hard board:

A screenshot of a computer

Description automatically generated

Very hard board:

A screenshot of a computer screen

Description automatically generated

1. The number of times the backtrack () function was called, and the number of times the backtrack () function returned failure, for each of the four boards.

Easy board:

* Backtrack count: 82
* Backtrack failure count: 0

Medium board:

* Backtrack count: 273
* Backtrack failure count: 191

Hard board:

* Backtrack count: 1288
* Backtrack failure count: 1206

Very hard board:

* Backtrack count: 14382
* Backtrack failure count: 14300

1. D and E are combined together as one answer
2. The runtime of the backtracking search algorithm for each of the four boards

The AC3 algorithm was run before backtracking. Since there was no requirement of implementing an inference function, the backtrack function I made has no form of inference and will therefore take an incredibly long amount of time to solve any of the boards.

* Easy board:
* Backtrack time: 7.65 milliseconds
* Total time taken for AC3 and backtrack: 14.19 milliseconds
* Medium board:
* Backtrack time: 11.02 milliseconds
* Total time taken for AC3 and backtrack: 20.74 milliseconds
* Hard board:
* Backtrack time: 102.17 milliseconds
* Total time taken for AC3 and backtrack: 107.18 milliseconds
* Very hard board:
* Backtrack time: 1109.64 milliseconds
* Total time taken for AC3 and backtrack: 1116.98 milliseconds

1. Brief discussion of your results. For example, explain why the AC-3 algorithm drastically decreases the total runtime when solving some Sudoku problems.

As stated in task D and E, The AC3 algorithm was run before backtracking. Without the AC3 algorithm the backtrack function takes too long to complete (I didn’t wait long enough to see the result because it took too long, I doubt it can even complete it in reasonable time), this is mostly because the backtrack function has no inference (a way to proactively reduce the search space). According to the task, it states “no inference needs to be performed during the backtracking search.”, therefore I did not add an inference in the backtracking search itself. But running the AC3 algorithm first to reduce the domains before running the backtrack function makes it possible for the backtrack function to find answers within a reasonable amount of time as shown in task D and E. The ordering of domain values and selecting unassigned variables methods can be improved. In the code’s current implementation, select\_unassigned\_variables() selects in the order the variables were initially added to the list. Similarly, the order of the domain values is not modified, and values are simply taken in the order they appear in the domain set.