Week 9 Project: Implementation FAQs

Bookmark this page

Q. My code return different answers between my local computer and Vocareum! What's the cause?

A. Please check in whether there is a Python version difference between Vocareum and your computer. As of July 2017, we use Python 3.4 on Vocareum for our grading environment. If you use Python 3.6 or newer locally, you might need to keep in mind the Dictionary ordering difference introduced in Python 3.6. One student reported sorting the resulted dictionary resolved the difference.

Q. What does it mean to solve a sudoku instance by AC-3 alone?

A. We consider a sudoku instance is solved if each of unassigned variables has only one domain value after AC-3.

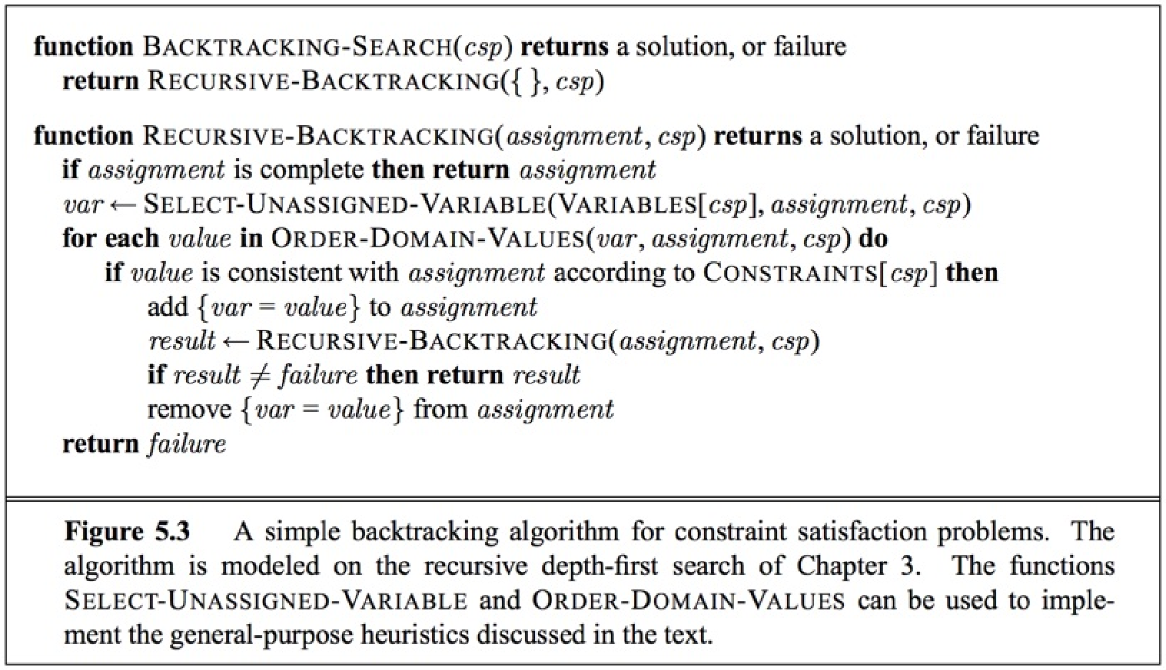
Q. My AC-3 has more than 1,000 constraints. It seems too much ... is this a correct approach?

A. Yes, it is a correct approach. Having thousands of constraints is usual in CSP solvers. If we express Sudoku's constraints in a concise way such as "each row/column/box must consist of all of the nine integers 1 through 9", it seems sudoku's constraints can be expressed in just this single constraint. However, AC-3 only recognizes constraints **in the form of arcs (binary relations)**, so we have to decompose this constraint into numerous binary constraints.

Q. In Backtracking Search pseudocode, why unassign() is not at the same indentation level as assign()? Should it be at the same level?

A. The pseudocode in class slides is taken from AIMA (Artificial Intelligence: Modern Approach) textbook, and the textbook's unassign() function essentially does no operation if **`var`** is not assigned, the pseudocode correctly works.

However, in some edition of the textbook, unassign() is placed at the same level as assign() as below:



```
function BACKTRACKING-SEARCH(csp) returns a solution, or failure
  return RECURSIVE-BACKTRACKING({ }, csp)

function RECURSIVE-BACKTRACKING(assignment, csp) returns a solution, or failure
  if assignment is complete then return assignment
  var ← SELECT-UNASSIGNED-VARIABLE(VARIABLES[csp], assignment, csp)
  for each value in ORDER-DOMAIN-VALUES(var, assignment, csp) do
     if value is consistent with assignment according to CONSTRAINTS[csp] then
        add {var = value} to assignment
        result ← RECURSIVE-BACKTRACKING(assignment, csp)
        if result ≠ failure then return result
        remove {var = value} from assignment
  return failure
```

Figure 5.3 A simple backtracking algorithm for constraint satisfaction problems. The algorithm is modeled on the recursive depth-first search of Chapter 3. The functions SELECT-UNASSIGNED-VARIABLE and ORDER-DOMAIN-VALUES can be used to implement the general-purpose heuristics discussed in the text.

(source: official AIMA website)

You can choose an appropriate indentation level depending on your preference.

English ▼