DQ6 (L6)

Due 5 Mar at 23:59 **Points** 25 **Questions** 12
Available after 13 Feb at 12:00 **Time limit** None
Allowed attempts Unlimited

Instructions

- This quiz is NOT GRADED. However, it is HIGHLY RECOMMENDED that you use these questions to complement your review of the lecture content.
- The questions are based on content from the Lecture 5b and Lecture 6, as well as from part of Chapter 5 of the AIMA (4th Ed.) textbook (i.e., 5.1-5.3).

Take the quiz again

Attempt history

	Attempt	Time	Score
KEPT	Attempt 2	5 minutes	25 out of 25
LATEST	Attempt 2	5 minutes	25 out of 25
	Attempt 1	16 minutes	21 out of 25

Submitted 6 Mar at 12:23

Question 1

2 / 2 pts

Which of the following are parts of a CSP formulation?

Correct!

- [x] Goal Test
- [ ] Evaluation Function

Correct!

- [x] Transition Model

Correct!

- [x] Actions

Correct!

- [x] State

Cost Function

We can disregard the evaluation function and the cost function when we formulate a search problem for CSP. However, we must still define the State, Actions and Transition via the variables in the CSP and their respective domains.

Question 2

2 / 2 pts

Consider the following statements.

Statement A: Formulating a constraint satisfaction problem (CSP) requires the definition of a set of variables, a set of domains (one for each variable), and a set of constraints (over the specified variables).

Statement B: For any CSP that does not contain global constraints, the vertices of its associated constraint graph correspond to variables of the problem, and the edges connecting vertices correspond to constraints of the problem.

Select the option that is true.

- Statement A is False; Statement B is False
- Statement A is False; Statement B is True
- Statement A is True; Statement B is False
- Correct! Statement A is True; Statement B is True

Both statements are true. Refer to lecture slides on CSPs.

Question 3

2 / 2 pts

Consider the following statements under the context of constraint satisfaction problems.

Statement A: Based on the backtracking algorithm specified in the lectures, given n variables (n > 0), there exist solutions at depth < n.

Statement B: A state in the CSP search tree corresponds to a partial assignment; such a partial assignment is one that only violates some (i.e., not all) of the constraints.

Select the option that is true.

Correct!

- Statement A is False; Statement B is False
- Statement A is False; Statement B is True
- Statement A is True; Statement B is False
- Statement A is True; Statement B is True

Statement A is False. Solutions are only found at depth n.

Statement B is False. A partial assignment is one that has some values being assigned to some variables in the CSP, ***with no violation of any constraints***. Note that an invariant of the CSP search tree is that that all nodes reference states that are consistent (i.e., we do not wish to explore inconsistent states since viable solutions cannot be found there - once a particular assignment to a variable causes a constraint to be inconsistent, that entire subtree must also be inconsistent, since we must continue the search with any assignment that has already been made). Consequently, each referenced state, i.e., each partial assignment must be consistent with ALL the constraints in the CSP.

Question 4

2 / 2 pts

Assume the use of the backtracking search algorithm discussed in the lectures on constraint satisfaction problems (CSP).

What is the maximum number of leaf nodes in a CSP Search Tree, given **6 variables**, and **5 values** in each domain?

- 7776
- Correct! 15625
- 9331200
- 11250000

Total number of leaves = d^n, where d is the number of values and n is the number of variables.

Note that we do not consider the order of assignments separately in the backtracking search algorithm.

Question 5

2 / 2 pts

Similar to binary constraints, unary constraints are checked for consistency throughout the backtracking search.

True or False?

- True
- Correct! False

False.

Unary constraints (i.e., node consistency) is checked as a pre-processing step prior to the execution of CSP backtracking search. Essentially, since, domains can only be reduced, once they are reduced sufficiently to satisfy the unary constraints, they would henceforth always satisfy the unary constraints (unless the domain becomes empty - at which point, we have reached a terminal state).

In other words, once you have removed all domain values that are inconsistent with the unary constraints, than any other value will be consistent with them, which means that no remaining domain values can invalidate them any longer. So, you are assured that any assignments made will be consistent with unary constraints.

Question 6

2 / 2 pts

Consider the following pseudocode for the CSP Backtracking algorithm.

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

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

Determine the heuristics that may be used to implement the following.

- A: SELECT-UNASSIGNED-VARIABLE(csp, assignment)

- B: ORDER-DOMAIN-VALUES(csp, var, assignment)

A: Minimum-remaining-values heuristic

B: Degree heuristic

Correct!

A: Minimum-remaining-values heuristic

B: Least-constraining-value heuristic

A: Degree Heuristic

B: Minimum-remaining-values heuristic

Correct!

A: Degree Heuristic

B: Least-constraining-value heuristic

A: Least-constraining-value heuristic

B: Minimum-remaining-values heuristic

A: Least-constraining-value heuristic

B: Degree Heuristic

MRV and degree heuristics are for variable order.

LCV is for value order.

Question 7

3 / 3 pts

During the lecture, the following three heuristics were described to perform variable and value-ordering for CSPs.

- Minimum-remaining-values (MRV) Heuristic
- Degree Heuristic
- Least-constraining-value (LCV) Heuristic

Consider the following options.

1. Variable

2. Value
3. For each variable x
4. For each unassigned variable, x
5. For each value in the domain of x
6. For each value in the consistent domain of x
7. Maximum
8. Minimum
9. Domain size for variable s
10. Consistent domain size for variable s
11. Number of constraints shared between s and other variables
12. Number of constraints shared between s and other unassigned variables
13. Sum of domain sizes across all variables that share a constraint with s
14. Sum of domain sizes across all unassigned variables that share a constraint with s
15. Sum of consistent domain sizes across all variables that share a constraint with s
16. Sum of consistent domain sizes across all unassigned variables that share a constraint with s

Complete the description for the following heuristics by selecting the correct options.

Minimum-remaining-values (MRV) Heuristic

- Selects [1] (Option 1/2) from the set S
- S is specified via the for-loop: [4] (Option 3/4/5/6)
- Selection of s ∈ S is based on [8] (Option 7/8) over [10] (Option 9/10/11/12)

Degree Heuristic

- Selects [1] (Option 1/2) from the set S
- S is specified via the for-loop: [4] (Option 3/4/5/6)
- Selection of s ∈ S is based on [7] (Option 7/8) over [12] (Option 9/10/11/12)

Least-constraining-value (LCV) Heuristic

- Selects [2] (Option 1/2) from the set S
- S is specified via the for-loop: [6] (Option 3/4/5/6)
- Selection of s ∈ S is based on [7] (Option 7/8) over [16] (Option 13/14/15/16)

Note that all blanks should only be filled with the integers from 1 to 16, each corresponding to the options listed above.

Answer 1:

Correct!

1

Answer 2:

Correct!

4

Answer 3:

Correct!

8

Answer 4:

Correct!

10

Answer 5:

Correct!

1

Answer 6:

Correct!

4

Answer 7:

Correct!

7

Answer 8:

Correct!

12

Answer 9:

Correct! 2

Answer 10:

Correct! 6

Answer 11:

Correct! 7

Answer 12:

Correct! 16

As specificed in the answer. Refer to lecture notes for examples.

Question 8

2 / 2 pts

In a general constraint satisfaction problem with n binary-valued variables, what is the worst-case complexity for the number of times the backtracking search will backtrack.

Choose the tightest bound.

- $O(1)$
- $O(n)$
- $O(n^2)$
- Correct! $O(2^n)$

In the worst case, we would traverse the whole search tree. Only in the final path considered (which leads to the goal), is there no backtracking.

Thus, we have 2^{n+1} - 1 - (n+1), or O(2^n).

Question 9

2 / 2 pts

The INFERENCE(csp, var, assignment) function within the BACKTRACKING-SEARCH algorithm may only be implemented using the AC-3 algorithm.

True or false?

The following is the pseudocode for the BACKTRACKING-SEARCH algorithm.

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

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

- True

Correct!

- False

False. We may also utilise forward checking as an inference algorithm.

Question 10

2 / 2 pts

Consider the following statements under the context of arc-consistency under constraint satisfaction problems.

Statement A: A binary constraint corresponds to 2 arcs.

Statement B: A variable X_i in a CSP is arc-consistent with respect to another variable X_j if and only if for every value $x \in D_i$, there exists some value $y \in D_j$ that satisfies the binary constraint on the arc (X_i, X_j).

Select the option that is true.

- Statement A is False; Statement B is False
- Statement A is False; Statement B is True
- Statement A is True; Statement B is False
- Correct! Statement A is True; Statement B is True

Both statements are true. Refer to lecture slides on CSPs.

Question 11

2 / 2 pts

The AC-3 algorithm is only used within the inference step of the CSP backtracking algorithm when solving a CSP.

True or False?

True

Correct!

False

False.

Inference algorithms are also often only used once as a pre-processing step to reduce domain sizes prior to the execution of the backtracking algorithm. For example, consider a Sudoku puzzle shown in the lecture; with such problems, the AC-3 algorithm may be employed to significantly reduce the domain of variables before the search for a solution commences.

Question 12

2 / 2 pts

Which of the following statements regarding the AC-3 algorithm are true?

function AC-3(csp) **returns** false if an inconsistency is found and true otherwise
    $queue \leftarrow$ a queue of arcs, initially all the arcs in csp

    while $queue$ is not empty **do**
        $(X_i, X_j) \leftarrow$ POP($queue$)
        if REVISE(csp, X_i, X_j) **then**
            if size of $D_i = 0$ **then return** $false$
            for each X_k **in** X_i.NEIGHBORS - $\{X_j\}$ **do**
                add (X_k, X_i) to $queue$
    return $true$

function REVISE(csp, X_i, X_j) **returns** true iff we revise the domain of X_i
    $revised \leftarrow false$
    for each x **in** D_i **do**
        if no value y in D_j allows (x,y) to satisfy the constraint between X_i and X_j **then**
            delete x from D_i
            $revised \leftarrow true$
    return $revised$

Each arc(X_i, X_j) is inserted **at least** d times (where d is the domain size for all variables).

Correct!

Each arc(X_i, X_j) is inserted **at most** d times (where d is the domain size for all variables).

A CSP has **at least** n^2 directed arcs.

Correct!

A CSP has **at most** n^2 directed arcs.

CSPs have ***at most*** 2 * (n choose 2) or $O(n^2)$ directed arcs (given n variables).

Each arc (X_i, X_j) can be inserted ***at most*** d times because X_i has at most d values to delete (given domain size d).