

Lecture 11 (Improving CSP)

1 Backtracking Search (with Inference)

```
function Backtracking(assignment, csp):
    if assignment is complete: return assignment
    var = Select-Unassigned-Variable(Variables(csp), assignment, csp)
    for each value in DomainValues(var, assignment, csp):
        if value is consistent with assignment given Constraints(csp):
            add {var = value} to assignment
            inferences = Inference(csp, var, assignment)
            if inferences != failure:
                add inferences to assignment
                result = Backtracking(assignment, csp)
                if result is not failure: return result
            remove {var = value} from assignment
    return failure
```

1.1 Inference - Forward Checking

Remove values of neighbours which are inconsistent with current assignment of node

2 Arc Consistency

1. Arc is consistent iff for any assignment for the source variable, there exists a valid assignment for the sink variable
2. The domain is modified accordingly by removing inconsistent values from the source

2.1 AC-3 Algorithm for Enforcing Arc Consistency

```
def AC-3(csp):
    queue = queue of all arcs in csp
    while queue is not empty:
        (Xi, Xj) = pop(queue)
        if revise(csp, Xi, Xj):
```

```

    if |Di| == 0:
        return False
    for each Xk in Xi.neighbours \ {Xj}:
        add (Xk, Xi) to queue
return True

```

Complexity of the algorithm is $O(n^2d^3)$. Backtracking with inference can also use AC-3 algorithm

2.2 Limitations

We cannot determine if a solution exists until we run a traversal on the graph, even after enforcing consistency

3 K-Consistency

1. For each k nodes, any consistent assignment to $k - 1$ can be extended to the k^{th} node
2. Arc consistency is special case for $k = 2$