CSE-4111 Artificial Intelligence Lab
Assignment 3: Arc Consistency in CSP
Saif Mahmud, Roll: SH-54

Problem Formulation

A binary constraint satisfaction problem (CSP) is a triple P = (X, D, C) where X = {1, ..., n} is a set of variables. Each variable $i \in X$ has a finite domain $D_i \in D$ of values that can be assigned to it. Arc consistency is defined as (i, a) is arc consistent with respect to the constraint C_{ij} if it is node consistent and there is a value $b \in D_j$ such that $(a, b) \in C_{ij}$. Such a value b is called support of a. Variable i is arc consistent if all its values are arc consistent with respect to every binary constraint involving i. A CSP is arc consistent (AC) if every variable is arc consistent. Arc inconsistent values can be removed because they cannot participate in any solution. AC is achieved by removing arc inconsistent values until a fixed point is reached. In this regard, there are 4 (four) algorithms attaining the goal of removing arc inconsistency from CSPs and they are namely AC-1, AC-2, AC-3, and AC-4.

Experiment Design

This experiment will incorporate the implementation of AC algorithms and measuring the performance of the algorithms against defined evaluation metrics.

Input

All four algorithms will take the CSP as the argument. Here, the nodes and edges of parameter CSP will be formulated with the following principles :

- Constraint graphs will be generated with a random number of nodes and edges. Here the nodes and edges will indicate variables and constraints respectively.
- Each variable or node will be assigned a set of discrete random values as a domain and not necessarily of the same size in case of every node.
- A pool of binary constraints will be created with a random logical relationship. Each edge in the CSP graph will be considered as the constraint between the two neighboring nodes. Each edge will randomly be assigned with a constraint pocked up from the predefined pool. It should be noted that the pool will be considerably large to ensure expected generalization of the constraint graphs.

Output
The implemented algorithms will return the graph holding all the constraints or in other words after elimination of inconsistency. If the resultant graph is a complete consistent graph then it has yielded a valid solution and otherwise the counterpart.

Performance Metric
The implementation of algorithms will be evaluated with the following metric in order to show comparison among them :
- CPU running time on the same input graph
- Change in performance with a gradual change in the size of input CSP graphs
- Time and Space Complexity of individual implementation of algorithms
- Mean constraint check and revision of nodes (in number) for each algorithm
- Performance on a well defined AI problem, for example, 8-Queen Problem