# CSC384
# Introduction to Artificial Intelligence:
# Backtracking Search

October 20, 2014

# Constraint Satisfaction Problems (CSPs)

## Recap

A CSP is a set of variables, domains and constraints:

- Variables: $V_1, ..., V_n$
- Variable Domains: $D_1, ..., D_n$
- Constraints: $C_1, ..., C_m$

We want to assign a value to each variable $V_i$ such that $v_i \in D_i$ and all constraints are satisfied.

- This is a variable assignment problem, the order in which assignments occur does not matter

# Constraint Satisfaction Problems (CSPs)

## Constraints

Constraints are defined over subset of the variables called its *scope*.

The size of a constraints scope is referred to as its *arity*.

- Constraints over a single variable are unary constraints.
- Constraints over two variables are binary constraints.
- Constraints over more variables are sometimes referred to as n-ary constraints.

An assignment of a subset of the variables is called a partial assignment. Constraints only restrict the partial assignments over its scope.

# Constraint Satisfaction Problems (CSPs)

## Constraints

Some common constraints:

- Table Constraints
- Inequalities (ex: $V_2 + 2V_4 - V_5 \geq V_7$)
- Alldifferent (All of the variables in the scope must have different values)
- Regular (defined by a finite automata)

Table constraints are the most basic constraint. It the standard format (a positive table) they list allowed partial assignments over the constraints scope. Individual table entries are sometimes refereed to as *tuples*.

Table constraints are called *extensional* constraints as they explicitly list allowed or disallowed tuples. Most other constraints are called *intensional* as the allowed or disallowed tuples are implicit

To solve a CSP we will implore a depth first search.

- We will update possible domain values under the current partial assignment
- If some domain has no possible values left (a domain wipeout) then we must backtrack
- On a backtrack we must restore the domain values to a previous state

We check what possible assignments can be next instead of making assignments until a conflict occurs. This is called *forward checking*.

We prune domain values to be consistent with constraints. Pruning one domain value may allow further pruning, this process is called *constraint propagation*.

# Constraint Satisfaction Problems (CSPs)

## Consistency

If domains under partial assignments conform to the constraints, they are consistent. Different forms of consistency refer to the amount of work done to ensure the constraints are obeyed.

*Node consistency*: all unary constraints are obeyed.

*Arc Consistency*: Variable $V_i$ is arc consistent with constraint $C_{ij}$ ($scope(C_{ij}) = \{V_i, V_j\}$) if $\forall a_i \in D_i \exists a_j \in d_j$ such that $(a_i, a_j) \in C_{ij}$

If we extend the definition of arc consistency to higher arity constraints we get *general arc consistency* or GAC.

# Constraint Satisfaction Problems (CSPs)

## AC3

Pseudocode for the basic arc consistency algorithm:

```
AC3(V,D,C)
for all (xi, xj) in constraint Cij
   queue.enqueue(xi,xj)
while queue is not empty
   (xi,xj) = queue.dequeue
   revise(xi,xj)
   if revise(xi,xj) caused a domain change in Di
      queue.enqueue(xi,xk) for all k != i or j
      such that Cik exists
```

This runs in $O(ek^3)$ time where $e$ is the number of binary constraints and $k$ is the maximum domain size.

# Constraint Satisfaction Problems (CSPs)

## AC4

We can do slightly better than AC3 by counting *supports*.

A support for a value $a_i \in D_i$ is a value in another domain $a_j \in D_j$ such that $(a_i, a_j) \in C_{ij}$. $a_j$ allows for an assignment of $a_i$ under the constraint $C_{ij}$.

When using table constraints we can count support for a domain value, pruning when the number of supports left drops to zero.

For intensional constraints finding a support may be more complicated. The complexity of this operation depends on how the constraint is implemented. Algorithms for handling different types of constraints are constraint specific.

This version runs in $O(ek^2)$ time.

# Constraint Satisfaction Problems (CSPs)

## Additional information

Other forms of consistency exist:

- Path consistency: consider multiple binary constraints
- K-consistency: generalization of arc and path consistency for $k > 3$
- Bounds consistency: consider only the bounds of the domains. Useful for fast execution under large domains.

This is not an exhaustive list!

# Constraint Satisfaction Problems (CSPs)

## Additional information

Some free available CSP solvers:

- Minion
- GECODE
- Mistral
- Choco