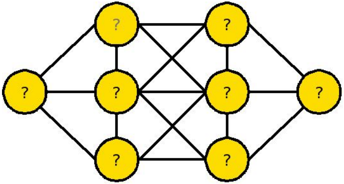
Constraints programming

Holy Grail of programming: the user states the problem, the computer solves it.

Example (Place numbers 1 through 8 on nodes)

1. each number appears exactly once
2. no connected nodes have consecutive numbers

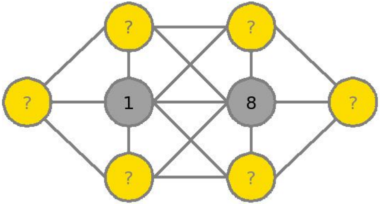


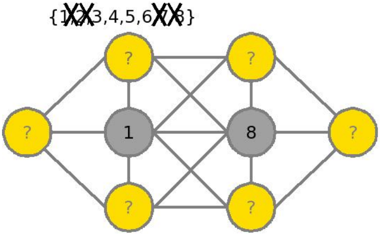
- Which nodes are hardest to number? (Guess a value, but be prepared to backtrack)
- Which are the least constraining values to use? (Symmetry means we don't need to consider: 8 1)

Heuristic Search

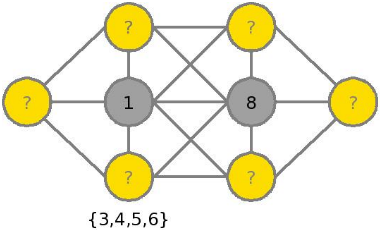
"To succeed, try first where you are most likely to fail."
"Deal with hard cases first: they can only get more difficult if you put them off."

Inference and constraint propagation I

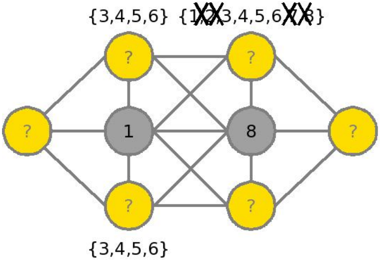




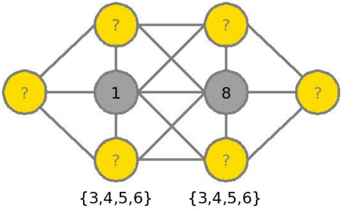
We can now eliminate many values for other nodes

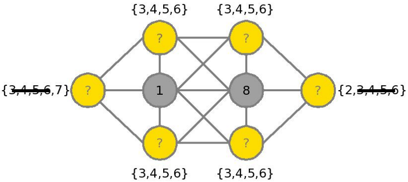


By simmetry

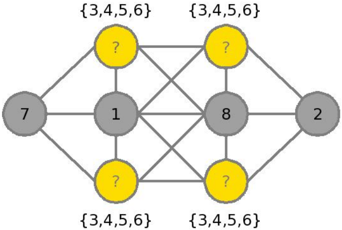


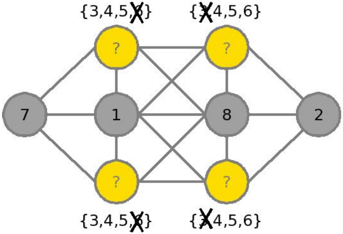
Inference & constraint propagation II





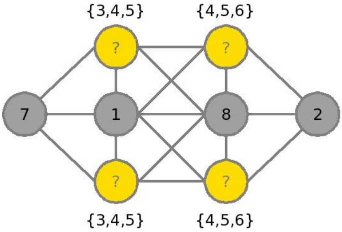
By symmetry

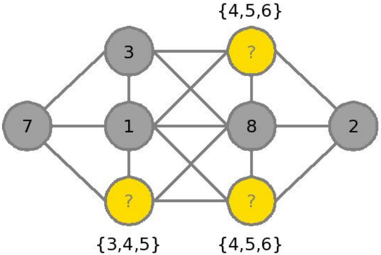




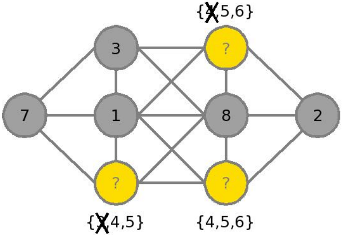
and propagate

Inference & constraint propagation III

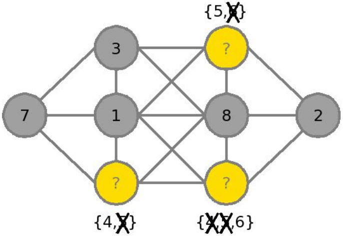




Guess a value, but be prepared to backtrack



More propagation?



Constraint programming methodology

1. Model problem
   - specify in terms of constraints on acceptable solutions:
   - define/choose constraint model: variables, domains, constraints
2. Solve model
   - define/choose algorithm
   - define/choose heuristics
3. Verify and analyze solution

Constraints Properties

A logical relation among several unknowns (variables)

- May specify partial information: $X > 2$, ”the circle is inside the square”
- Non-directional: two variables X, Y can be used to infer a constraint on X given a constraint on Y and vice versa: X=Y+2
- Declarative: specify what relationship must hold without specifying a computational procedure to enforce that relationship
- Additive: the order of imposition of constraints does not matter, all that matters, at the end is that the conjunction of constraints is in effect
- Rarely independent: typically constraints in the constraint store share variables.

Constraint satisfaction problem (CSP)

A CSP is defined by

- a set of variables: X, Y, Z,
- a domain of values for each var: $X : \{1, 2\}$, $Y : \{1, 2\}$, $Z : \{1, 2\}$
- a set of constraints between variables: $X = Y$, $X \neq Z$, $Y > Z$

A solution is an assignment of a value to each variable that

satisfies the constraints: $X = 2$, $Y = 2$, $Z = 1$

Given a CSP

- Determine whether it has a solution or not (satisfiability)
- Find any solution
- Find all solutions
- Find an optimal solution, given some cost function

Constraint model for 8 digit puzzle

1. each number appears exactly once
2. no connected nodes have consecutive numbers

```
set(arithmetic).
assign(domain_size,9).
assign(max_models,-1).

list(distinct).
 [0,a,b,c,d,e,f,g,h].
end_of_list.

formulas(assumptions).
  abs(a + (-b)) != 1.   abs(a + (-g)) != 1.   abs(a + (-f)) != 1
  abs(b + (-c)) != 1.   abs(b + (-h)) != 1.   abs(b + (-g)) != 1
  abs(c + (-d)) != 1.   abs(c + (-h)) != 1.   abs(c + (-g)) != 1
  abs(d + (-h)) != 1.   abs(d + (-e)) != 1.
  abs(e + (-f)) != 1.   abs(e + (-g)) != 1.   abs(e + (-h)) != 1
  abs(f + (-g)) != 1.   abs(f + (-h)) != 1.
  abs(g + (-h)) != 1.
end_of_list.
```

Variables: $a, \ldots, h$

Domains: $\{1, \ldots, 8\}$

Constraints: $|a - b| \neq 1 \ldots$
$alldifferent(a, \ldots, h)$