CS 316: Constraint satisfaction problems

Stefan D. Bruda

Winter 2023

CONSTRAINT SATISFACTION PROBLEMS

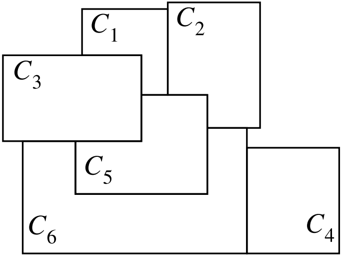
- Standard search problem: the state is anything that supports goal test, comparison, successor
- CSP: the state is defined by variables V_i with values from domains D_i
  - The goal test is a set of constraints, which specifies allowable combinations of values for subsets of variables. A state is a set of variable bindings.

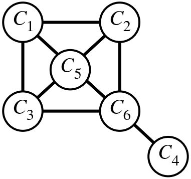
$$
\begin{array}{rll}
\mathit{Variables} & \{ & \mathit{Shoes}, \mathit{Pants}, \mathit{Shirt} \quad \} \\
\mathit{Domains} & \{ & \{\mathit{Sandals}, \mathit{Runners}\}, \{\mathit{Jeans}, \mathit{Blue}, \mathit{Grey}\}, \{\mathit{Green}, \mathit{White}\} \quad \} \\
\mathit{Constraints} & \{ & (\mathit{Shoes} = \mathit{Sandals}, \mathit{Pants} = \mathit{Grey}), \\
 & & (\mathit{Shoes} = \mathit{Runners}, \mathit{Pants} = \mathit{Jeans}), \\
 & & (\mathit{Shoes} = \mathit{Sandals}, \mathit{Shirt} = \mathit{Green}), \\
 & & (\mathit{Shoes} = \mathit{Runners}, \mathit{Shirt} = \mathit{White}), \\
 & & (\mathit{Pants} = \mathit{Grey}, \mathit{Shirt} = \mathit{Green}), \\
 & & (\mathit{Pants} = \mathit{Jeans}, \mathit{Shirt} = \mathit{White}), \\
 & & (\mathit{Pants} = \mathit{Blue}, \mathit{Shirt} = \mathit{White}) \quad \}
\end{array}
$$

  - This is actually an example of binary CSP

Example: Map Colouring

- Colour a map so that no adjacent countries have the same colour.





- Variables: Countries C_i
- Domains: $\{Red, Green, Blue\}$
- Constraints: $C_1 \neq C_2$, $C_1 \neq C_3$, $C_3 \neq C_5$, ...

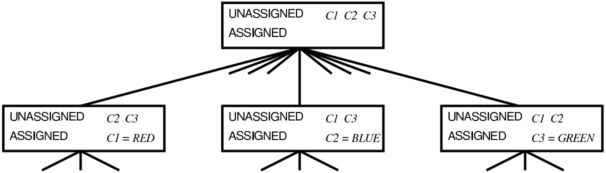
Real-world CSP

- Assignment problems
    - e.g., who teaches what class
- Timetabling problems
    - e.g., which class is offered when and where?
- Hardware configuration
- Spreadsheets
- Transportation scheduling
- Factory scheduling

Solving CSP by standard search methods

- States are defined by the variables bound so far.

Initial state	All variables unbound.
Operators	Bind one variable
Goal test	All variables assigned, no constraints violated



- Disadvantages?

IMPROVING THE CSP ALGORITHM

- Order of assignment is irrelevant (many paths are equivalent)
- Further bindings cannot correct an already violated constraint
- We can use depth-first search, but
    - Fix the order of assignment
    - Check for constraint violations
        - at the SUCCESSORS level, or immediately before expanding the state.

```
function CSP-SEARCH() returns a solution, or failure

nodes ← MAKE-QUEUE(MAKE-NODE(INITIAL-STATE))
repeat
   if nodes is empty then return failure
   node ← REMOVE-FRONT(nodes)
   if GOAL-TEST(node) then return node
   unless VIOLATES-CONSTRAINTS(node) do
      nodes ← APPEND(SUCCESSORS(node), nodes)
forever
```

- We do not need the queue actually (why?)

Backtracking

- We fix the order of assignment, and we check for constraint violations
- The resulting algorithm is called backtracking, the basic uninformed algorithm for CSP. Can solve *n*-queens for $n \approx 15$

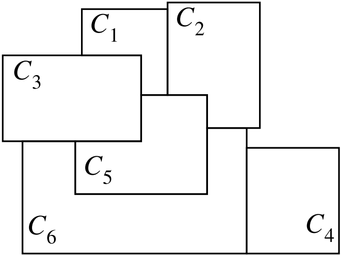
```
function Backtracking(state, variables, domains) returns a solution,
or failure

var ← First(variables)
domain ← First(domains)
foreach val in domain do
   unless Violates-Constraints(Add((var,val),state)) do
      if no more variables then return state
      Backtracking(Add((var, val), state),
                   Rest(variables), Rest(domains))
end
return failure
```

Backtracking improvements

- Variant: forward checking, looks ahead and erases from the domains of all the variables those values that cannot be assigned without violating constraints
    - Forward checking is a particular case of arc consistency working on the graph generates by the constraints
    - Arc consistency can be applied on the initial graph of constraints before performing the backtracking search

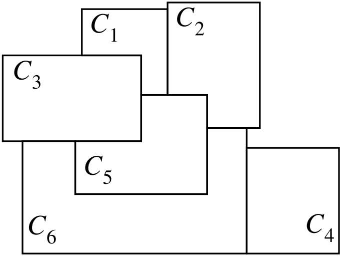
Heuristics for CSP

- We can make more intelligent decisions on
  - which value to choose for each variable
  - which variable to assign next
- Given $C_1 = \textit{Red}$ and $C_2 = \textit{Green}$, $C_3 =$?
- Given $C_1 = \textit{Red}$, $C_2 = \textit{Green}$, what next?



Heuristics for CSP

- We can make more intelligent decisions on
  - which value to choose for each variable
  - which variable to assign next
- Given $C_1 = \textit{Red}$ and $C_2 = \textit{Green}$, $C_3 = ?$
  - $C_3 = \textit{Green}$, the least constraining value
- Given $C_1 = \textit{Red}$, $C_2 = \textit{Green}$, what next?
  - Choose C_5, the most constrained variable



- Can solve n-queens for $n \approx 1000$