Constraint Satisfaction Problems/Programming

A4B33ZUI, LS 2016

Branislav Bošanský, Ondřej Vaněk, Štěpán Kopřiva

{name.surname}@agents.fel.cvut.cz

Artificial Intelligence Center, Czech Technical University

State space search → CSP

- until now – no assumptions on the states
- CSP:
  - state → array of variables
  - context → domains for variables, set of constraints among the variables
  - goal → all variables have an assigned value, no constraint is violated
- alternative goals
  - optimization variant

Advantages/Disadvantages

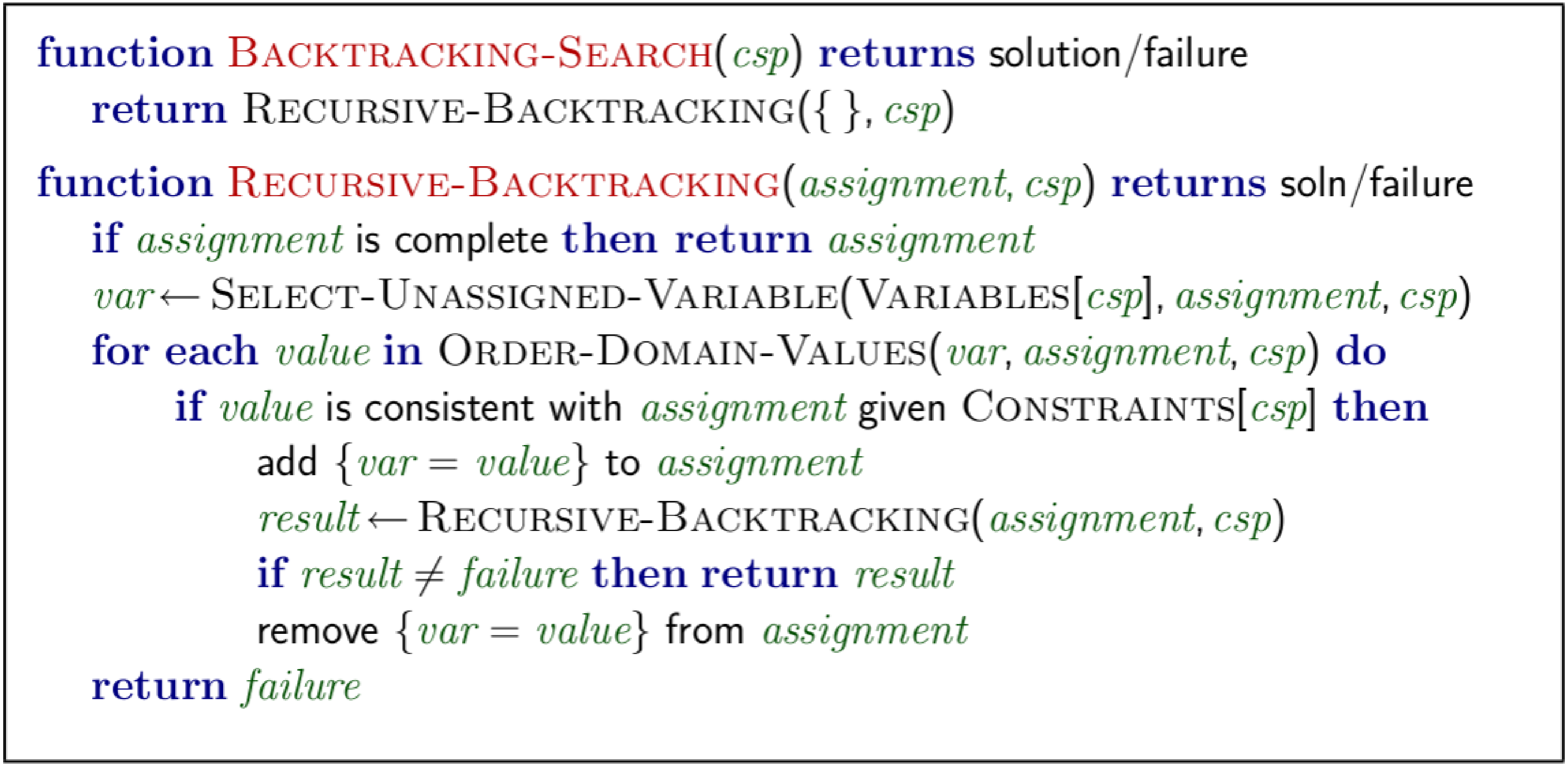
- general enough to model many problems
- more efficient algorithms that exploit the structure
  - we can do better than DFS/BFS
  - advanced search techniques can be possibly reused in non-CSP problems
  - generic purpose CSP solvers
- not all problems can be modeled as CSP
- some formulations can be inefficient

Example

- sudoku

- scheduling requests to hotel rooms
  - list of requests (# of people, from, to)
  - set of rooms (# of beds)
- scheme of the algorithm?
  - states / actions / search algorithm?

Example



```
function BACKTRACKING-SEARCH($csp$) returns solution/failure
    return RECURSIVE-BACKTRACKING({ }, $csp$)

function RECURSIVE-BACKTRACKING($assignment, csp$) returns soln/failure
    if $assignment$ is complete then return $assignment$
    $var$ ← SELECT-UNASSIGNED-VARIABLE(VARIABLES[$csp$], $assignment, csp$)
    for each $value$ in ORDER-DOMAIN-VALUES($var, assignment, csp$) do
        if $value$ is consistent with $assignment$ given CONSTRAINTS[$csp$] then
            add $\{var = value\}$ to $assignment$
            $result$ ← RECURSIVE-BACKTRACKING($assignment, csp$)
            if $result \neq failure$ then return $result$
            remove $\{var = value\}$ from $assignment$
    return $failure$
```

Backjumping

Conflict-driven backjumping

Constraints Graph

Dynamic backjumping

Uzly obarvit barvami 1,2,3, poradi A, B, C, D, E