CS370

Symbolic Programming
Declarative Programming

LECTURE 14: Problem Decomposition and AND/OR Graphs

Jong C. Park

park@cs.kaist.ac.kr

Computer Science Department

Korea Advanced Institute of Science and Technology

http://nlp.kaist.ac.kr/~cs370

Problem Decomposition and AND/OR Graphs

- AND/OR graph representation of problems
- Examples of AND/OR representation
- Basic AND/OR search procedures
- Best-first AND/OR search

AND/OR graph representation

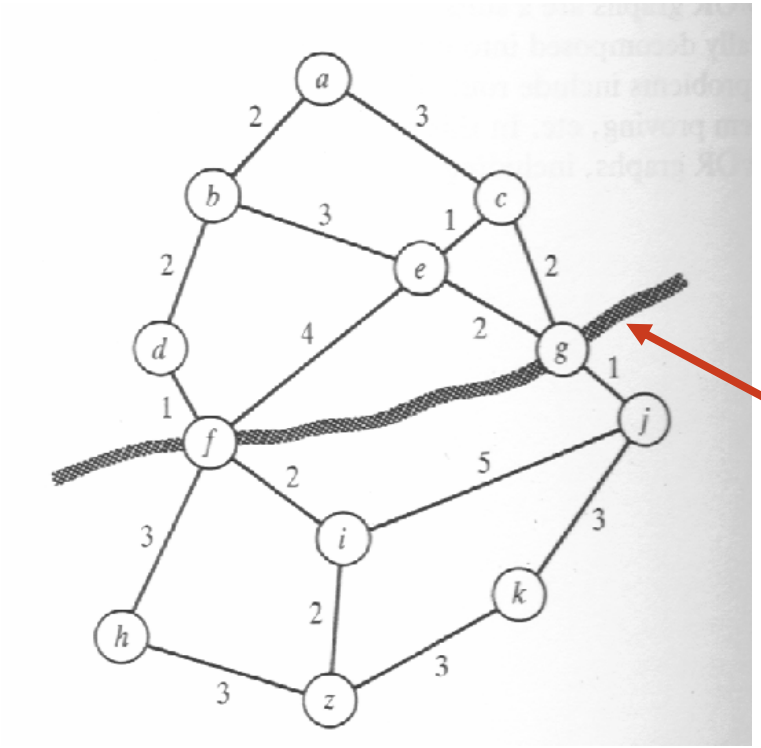
Areas of use

- Route Finding
- Design
- Symbolic Integration
- Game Playing
- Theorem Proving

AND/OR graph representation

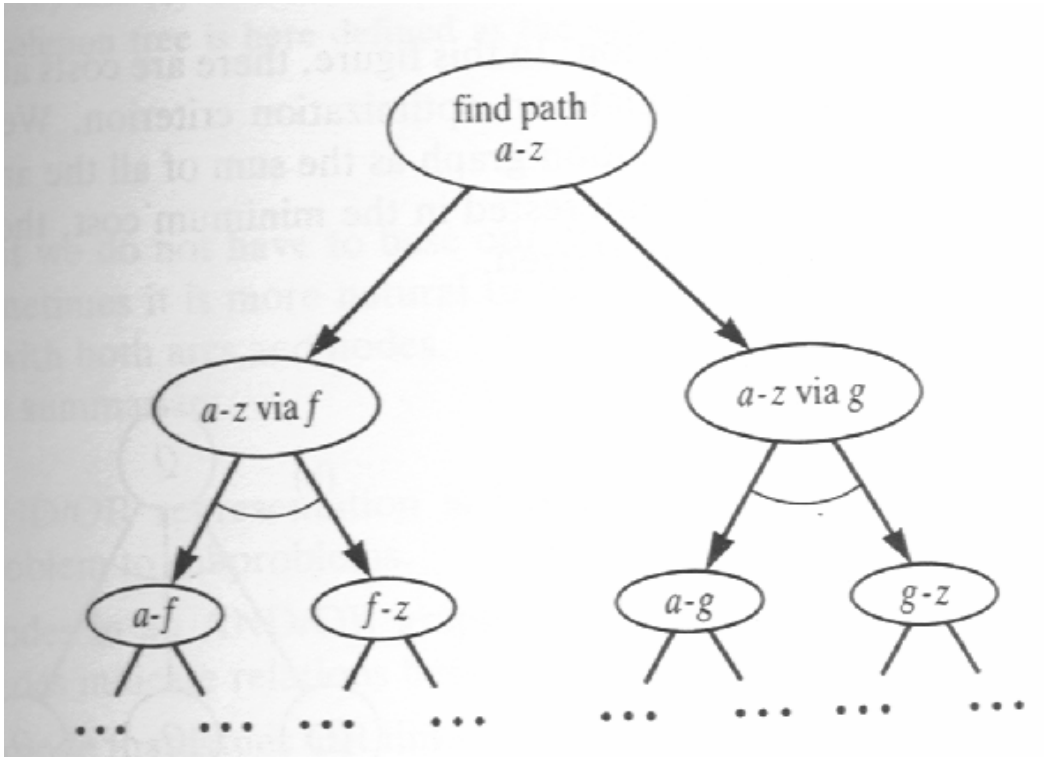
- **Finding a route from a to z in a road map**

To find a path between a and z, find *either*
(1) a path from a to z
via f, *or*
(2) a path from a to z
via g.



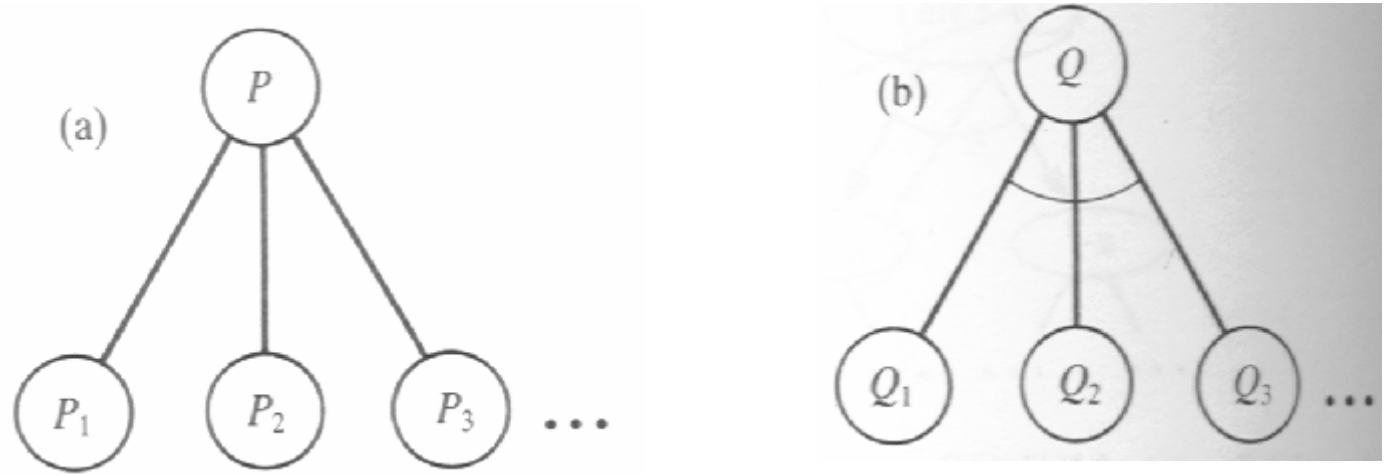
AND/OR graph representation

⊙ An AND/OR representation



AND/OR graph representation

⊙OR and AND relations



to solve P, solve any of P_1 or P_2 or ... to solve Q, solve all Q_1 and Q_2 ...

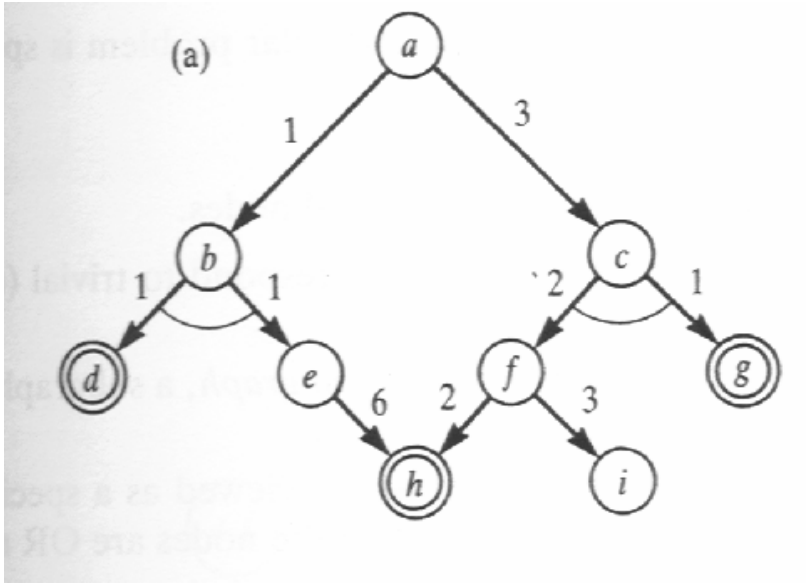
AND/OR graph representation

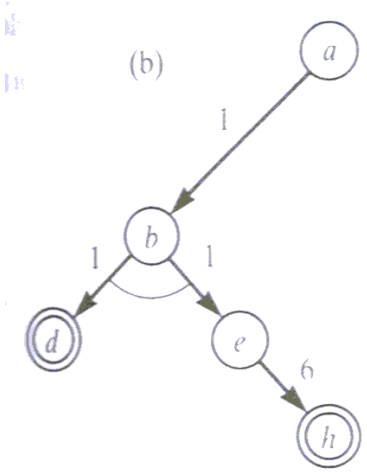
⊙Solution

- In the state-space representation, a solution to the problem is a path in the state space.
- In the AND/OR graph representation, a solution is a tree that includes all the subproblems of an AND node.

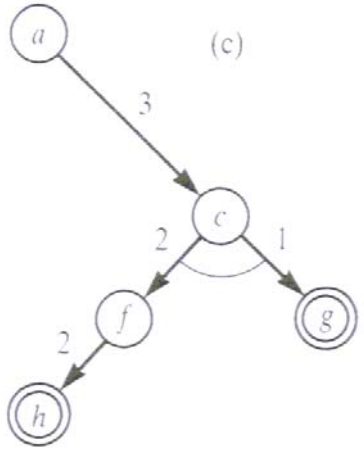
AND/OR graph representation

- An AND/OR graph and two solution trees





cost = 9



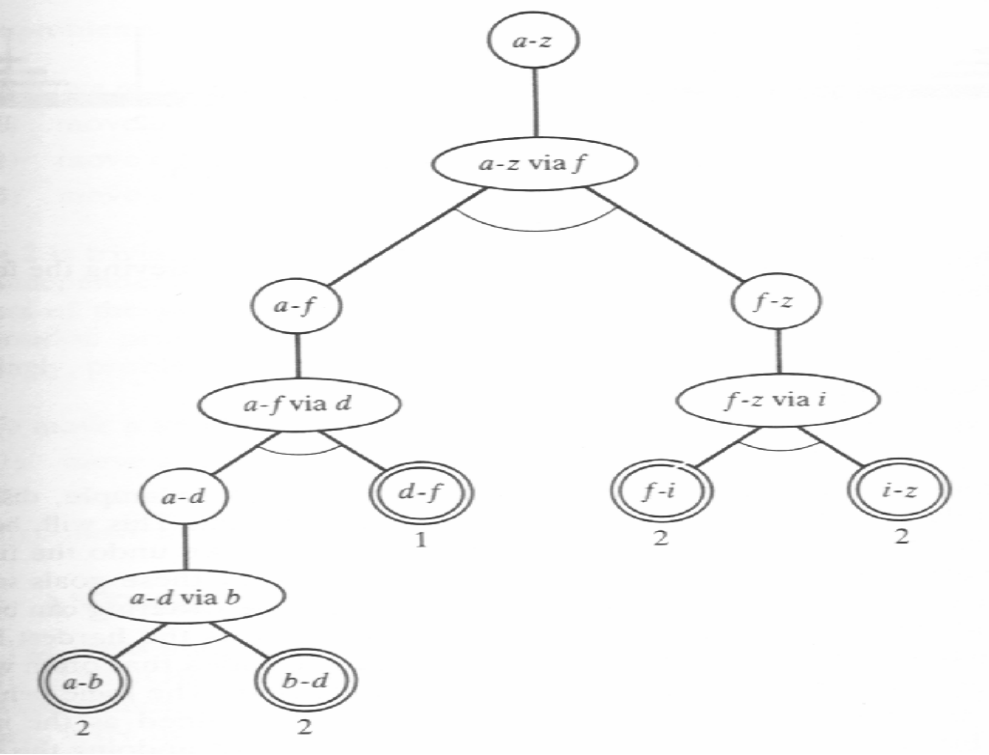
cost = 8

Examples of AND/OR representation

- **AND/OR representation of route finding**
  - OR nodes of the form X-Z
  - AND nodes of the form X-Z via Y
  - A goal node X-Z
  - The cost of each goal node X-Z
  - The costs of all other (non-terminal) nodes

Examples of AND / OR representation

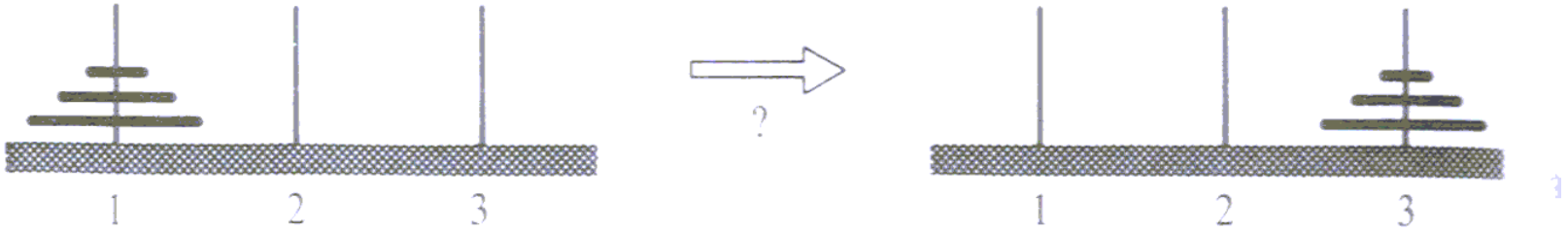
- AND / OR representation of route finding



[a,b,d,f,i,z]

Examples of AND/OR representation

⊙The Tower of Hanoi problem



- the set of goals
  - Disk a on peg 3
  - Disk b on peg 3
  - Disk c on peg 3

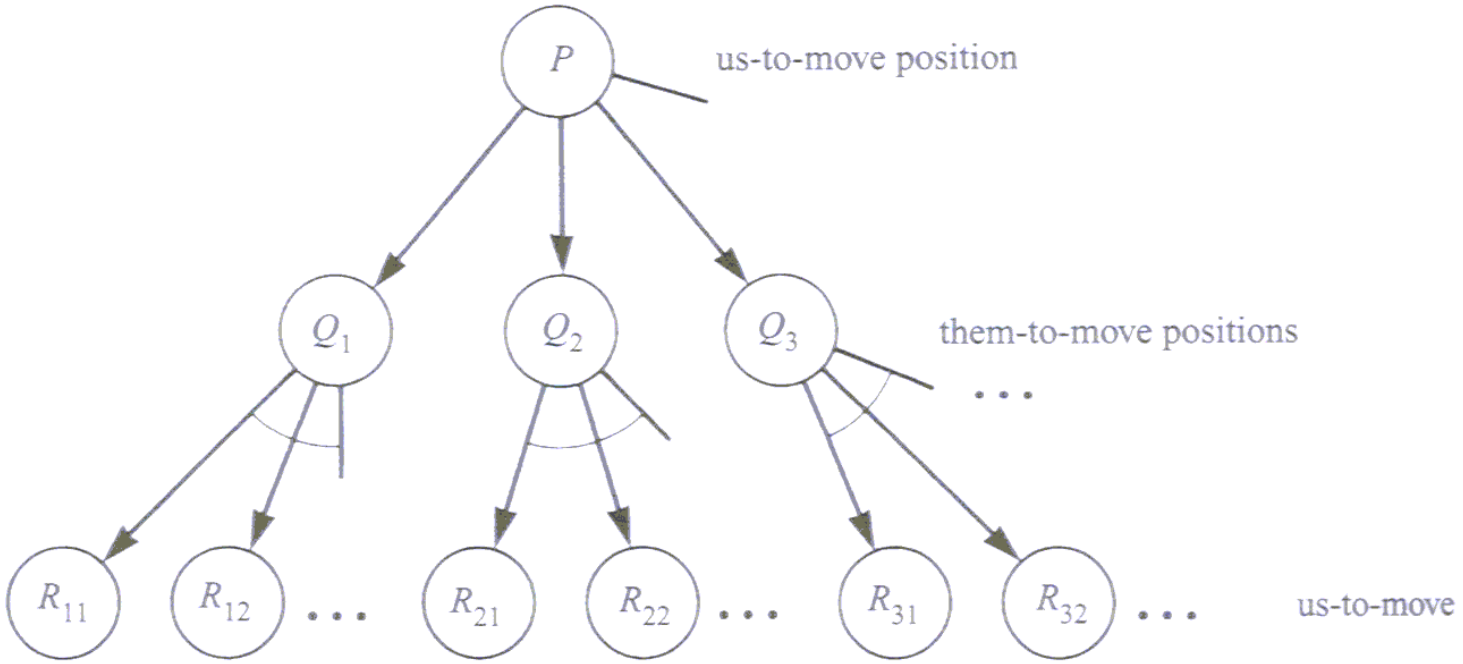
Examples of AND/OR representation

⊙ The Tower of Hanoi problem

- Refined strategy
  - Enable moving disk c from 1 to 3
  - Move disk c from 1 to 3
  - Achieve remaining goals: a on 3, and b on 3
- How?

Examples of AND/OR representation

⊙AND/OR formulation of game playing



Basic AND/OR search procedures

Searching AND/OR graphs in Prolog

- Use Prolog's own search mechanism

```
a :- b.
a :- c.
b :- d, e.
e :- h.
c :- f, g.
f :- h; i.
d.
g.
h.
?- a.
```

Basic AND/OR search procedures

⊙Searching AND/OR graphs in Prolog

- Use Prolog's own search mechanism
  - Problems
    - producing a solution tree
    - handling costs
    - dealing with cycles

Basic AND/OR search procedures

⊙Searching AND/OR graphs in Prolog

- Introducing the operator '--->'

```
:- op(600, xfx, --->).
:- op(500, xfx, :).
a ---> or:[b,c].
b ---> and:[d,e].
c ---> and:[f,g].
e ---> or:[h].
f ---> or:[h,i].
goal(d).          goal(g).          goal(h).
```

Basic AND/OR search procedures

⊙Searching AND/OR graphs in Prolog

- the depth-first AND/OR procedure

```
solve(Node) :- goal(Node).
solve(Node) :- Node ---> or:Nodes,
               member(Node1,Nodes),
               solve(Node1).
solve(Node) :- Node ---> and:Nodes,
  solveall(Nodes).
solveall([ ]).
solveall([Node|Nodes]) :- solve(Node),
  solveall(Nodes).
```

Basic AND/OR search procedures

⊙Searching AND/OR graphs in Prolog

- producing a solution tree

```
solve(Node, Node) :- goal(Node).
solve(Node, Node ---> Tree) :-
                    Node ---> or:Nodes,
                    member(Node1,Nodes),
                    solve(Node1, Tree).
solve(Node, Node ---> and:Trees) :-
  Node ---> and:Nodes, solveall(Nodes, Trees).
solveall([ ], [ ]).
solveall([Node|Nodes], [Tree|Trees]) :-
  solve(Node, Tree), solveall(Nodes, Trees).
```

Basic AND/OR search procedures

⊙ Searching AND/OR graphs in Prolog

- limiting the depth of search

```
solve(Node, Node, MaxDepth) :- goal(Node).
solve(Node, Node ---> Tree, MaxDepth) :-
  MaxDepth > 0, Node ---> or:Nodes,
  member(Node1,Nodes), Depth1 is MaxDepth - 1,
  solve(Node1, Tree, Depth1).
solve(Node, Node ---> and:Trees, MaxDepth) :-
  MaxDepth > 0, Node ---> and:Nodes,
  Depth1 is MaxDepth - 1,
  solveall(Nodes, Trees, Depth1).
```

Basic AND/OR search procedures

⊙Searching AND/OR graphs in Prolog

- Iterative deepening

```
iterative_deepening(Node, SolTree) :-
  trydepths(Node, SolTree, 0).
trydepths(Node, SolTree, Depth) :-
  solve(Node, SolTree, Depth)
  ;
  Depth1 is Depth + 1,
  trydepths(Node, SolTree, Depth1).
```

Best-first AND/OR search

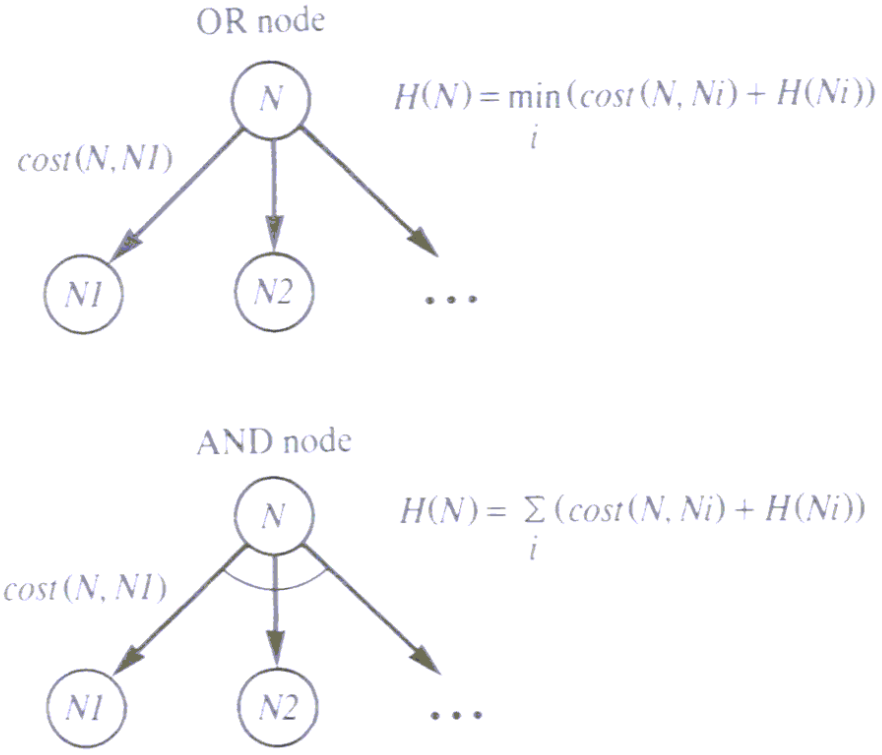
⊙ Heuristic estimates & the search algorithm

- Extending the AND/OR representation

```
a ---> or:[b/1,c/3].
b ---> and:[d/1,e/1].
c ---> and:[f/2,g/1].
e ---> or:[h/6].
f ---> or:[h/2,i/3].
goal(d).
goal(g).
goal(h).
```

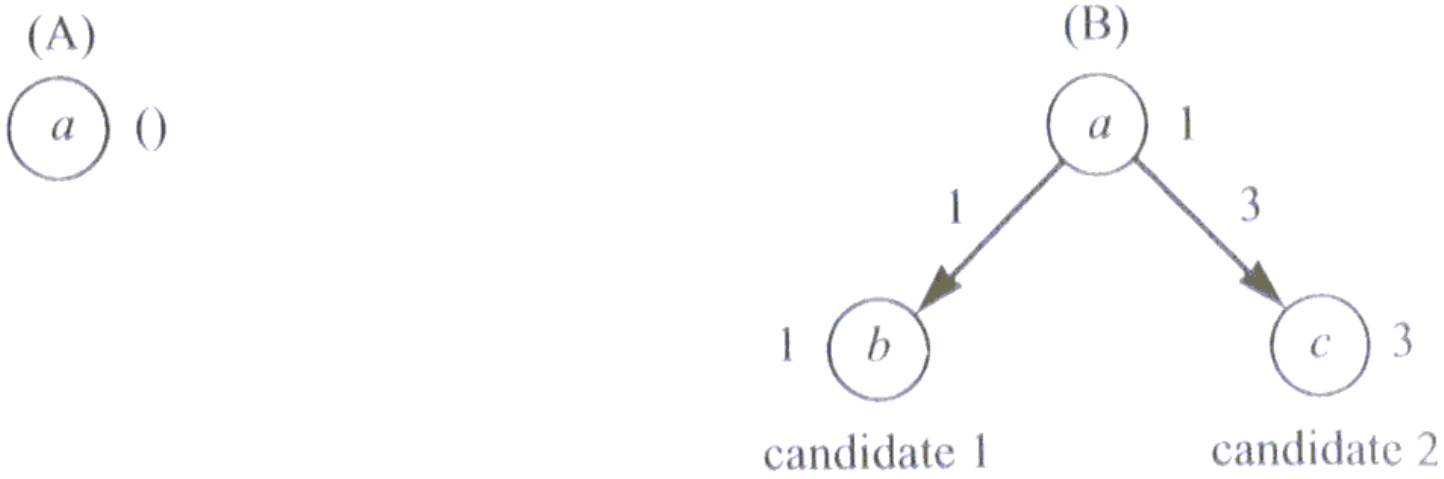
Best-first AND/OR search

- Heuristic estimates & the search algorithm



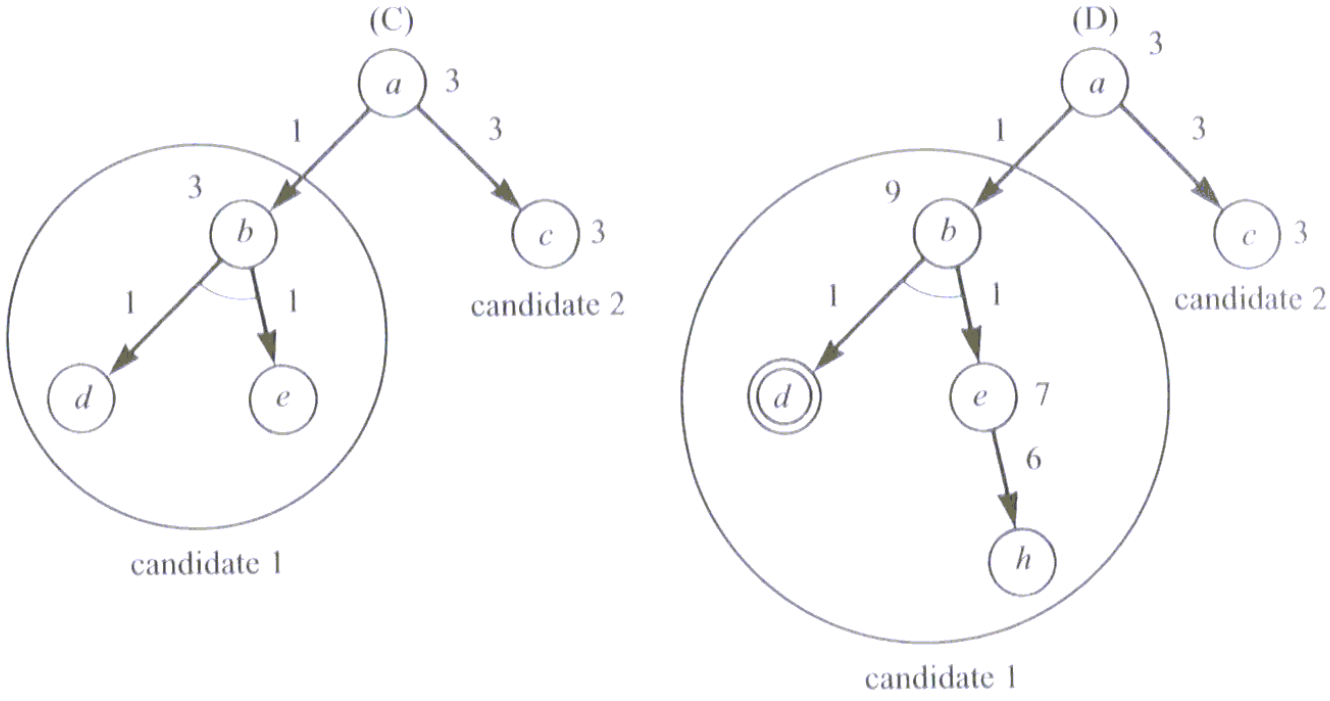
Best-first AND/OR search

- Search with F(N) = cost(M,N) + H(N)



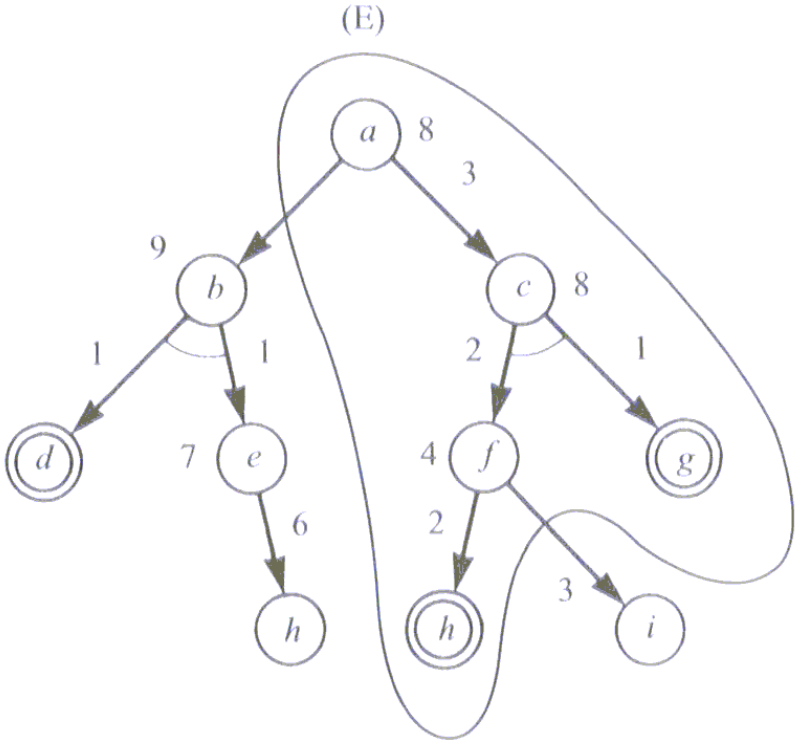
Best-first AND/OR search

⊙Search with F(N) = cost(M,N) + H(N)



Best-first AND/OR search

⊙Search with F(N) = cost(M,N) + H(N)



Best-first AND/OR search

⊙Example: route finding

- the road map: s(City1,City2,D)
- key point: key(City1-City2,City3)

  key(a-z,f).

  key(a-z,g).
- two kinds of problems
  - X-Z: find a route from X to Z
  - X-Z via Y: find a route from X to Z through Y

Best-first AND/OR search

⊙Example: route finding

```
:- op(560, xfx, via).
X-Z --> or:ProblemList :-
  bagof((X-Z via Y)/0,key(X-Z,Y),ProblemList), !.
X-Z --> or:ProblemList :-
  bagof((Y-Z)/D,s(X,Y,D),ProblemList).
X-Z via Y --> and:[(X-Y)/0,(Y-Z)/0].
goal(X-X).
```

Summary

- AND/OR graph representation of problems
- Examples of AND/OR representation
- Basic AND/OR search procedures
- Best-first AND/OR search