UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Workbook: Adversarial Search

Albert Sanchis

DSIC

Departament de Sistemes Informàtics i Computació

Learning objectives

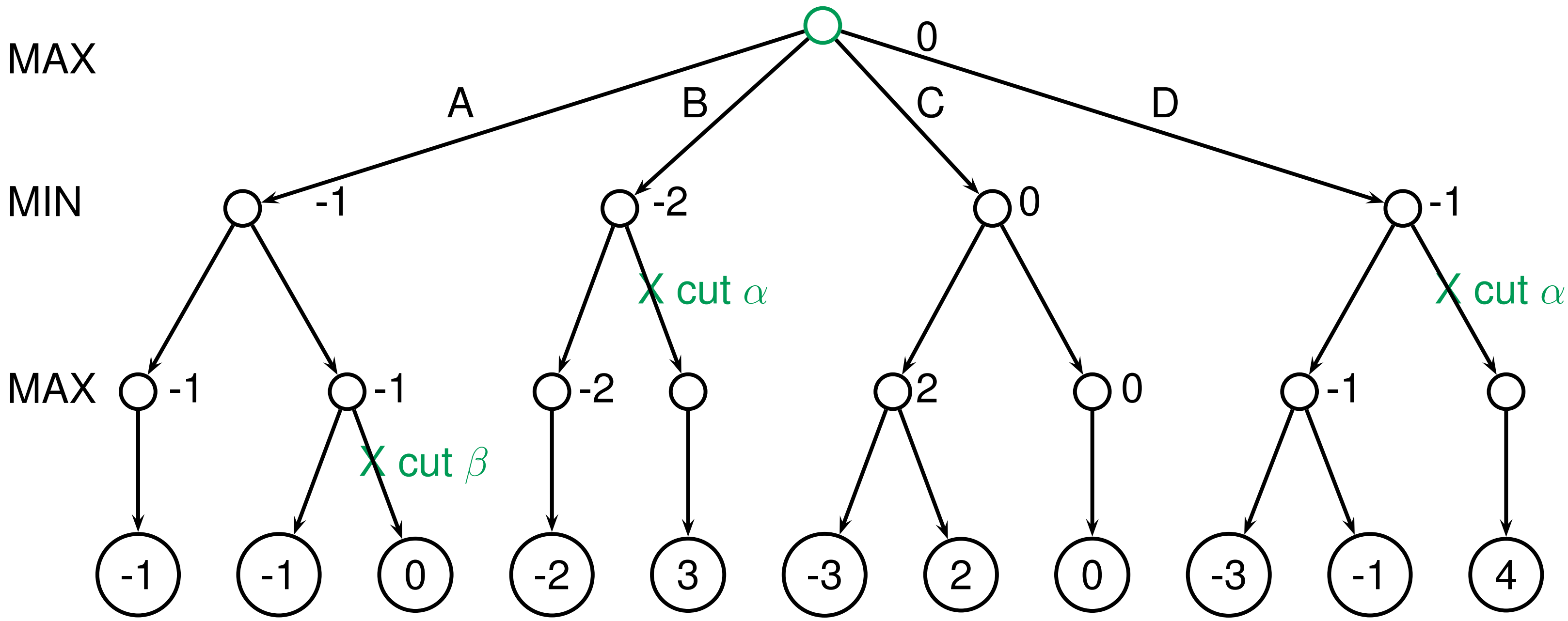
- To know the basics of adversarial search.
- To apply the ***minimax*** algorithm and ***alpha-beta*** pruning.

Minimax algorithm and alpha-beta pruning

```
mm(s, d, max)                              // state, depth, max="Does MAX move?"
  if  s is terminal:  return  utility for s
  if  d = 0:           return  heuristic value for s
  if  max: v = −∞;  ∀ n ∈ succ(s):  v = max(v, mm(n, d − 1, FALSE))
  else:     v = ∞;   ∀ n ∈ succ(s):  v = min(v, mm(n, d − 1, TRUE))
  return  v
```

```
α-β(s, d, α, β, max)
  if  s is terminal:  return  utility for s
  if  d = 0:           return  heuristic value for s
  if  max:  v = −∞
            ∀ n ∈ succ(s)
                v = max(v, α-β(n, d − 1, α, β, FALSE))
                α = max(α, v);   if β ≤ α: break   // β cut
  else:     v = ∞
            ∀ n ∈ succ(s)
                v = min(v, α-β(n, d − 1, α, β, TRUE))
                β = min(β, v);   if β ≤ α: break   // α cut
  return  v
```

- ***Question 1***: Draw the search tree of applying the ***minimax and alpha-beta pruning*** algorithm to the search space of a game shown in the following figure:



- ***Question 2***: What is the best move from the root node applying the ***minimax and alpha-beta pruning*** algorithm to the previous search space? ***Branch C***

- ***Question 3***: How many nodes are not generated applying the ***minimax and alpha-beta pruning*** algorithm instead of the basic minimax algorithm? ***5 nodes***

- ***Question 4***: What value should store the "question" node so that the cut does not happen? ***A value greater than zero***