

Algoritmos estándar de búsqueda con adversario

```
function minimax(nodo, profundidad, deboMaximizar)
  if profundidad = 0 or nodo es terminal
    return evaluación heurística del nodo
  if deboMaximizar
     $\alpha := -\infty$ 
    for each siguiente acción posible
       $\alpha := \max(\alpha, \text{minimax}(\text{hijo}, \text{profundidad} - 1, \text{False}))$ 
    return  $\alpha$ 
  else
     $\beta := +\infty$ ,
    for each siguiente acción posible
       $\beta := \min(\beta, \text{minimax}(\text{hijo}, \text{profundidad} - 1, \text{True}))$ 
    return  $\beta$ 

(* Llamada inicial *)
minimax(origen, profundidad, True)
```

```
function alphabeta(nodo, profundidad,  $\alpha$ ,  $\beta$ , deboMaximizar)
  if profundidad = 0 or nodo es terminal
    return evaluación heurística del nodo
  if deboMaximizar
    for each siguiente acción posible
       $\alpha := \max(\alpha, \text{alphabeta}(\text{hijo}, \text{profundidad} - 1, \alpha, \beta, \text{False}))$ 
      if  $\beta \leq \alpha$ 
        break
    return  $\alpha$ 
  else
    for each siguiente acción posible
       $\beta := \min(\beta, \text{alphabeta}(\text{hijo}, \text{profundidad} - 1, \alpha, \beta, \text{True}))$ 
      if  $\beta \leq \alpha$ 
        break
    return  $\beta$ 

(* Llamada inicial *)
alphabeta(origen, profundidad,  $-\infty$ ,  $+\infty$ , True)
```

```
function expectimax(nodo, profundidad, deboMaximizar)
  if profundidad = 0 or nodo es terminal
    return evaluación heurística del nodo
  if deboMaximizar
     $\alpha := -\infty$ 
    for each siguiente acción posible
       $\alpha := \max(\alpha, \text{expectimax}(\text{hijo}, \text{profundidad} - 1, \text{False}))$ 
    return  $\alpha$ 
  else
     $\beta := 0$ 
    numAcciones = 0
    for each siguiente acción posible
      numAcciones += 1
       $\beta := \beta + \text{expectimax}(\text{hijo}, \text{profundidad} - 1, \text{True})$ 
    return  $\beta/\text{numAcciones}$ ;
```

```
(* Llamada inicial *)  
expectimax(origen, profundidad, True)
```