

Minimax Implementation (Dispatch)

def value(state):

if the state is a terminal state: return the state's utility

if the next agent is **MAX**: return **max-value(state)**

if the next agent is **MIN**: return **min-value(state)**

def max-value(state):

initialize $v = -\infty$

for each successor of state:

$v = \max(v, \text{value}(\text{successor}))$

return v

def min-value(state):

initialize $v = +\infty$

for each successor of state:

$v = \min(v, \text{value}(\text{successor}))$

return v

Minimax algorithm

Depth first search

```
function MINIMAX_DECISION(state) returns an action  
return  $\max_{a \in \text{ACTIONS}(state)} \text{MIN\_VALUE}(\text{RESULT}(state, a))$ 
```

```
function MAX_VALUE(state) returns a utility value  
if TERMINAL_TEST(state) then return UTILITY(state)  
 $v \leftarrow -\infty$   
for each  $a$  in ACTIONS(state) do  
     $v \leftarrow \text{MAX}(v, \text{MIN\_VALUE}(\text{RESULTS}(state, a)))$   
return  $v$ 
```

```
function MIN_VALUE(state) returns a utility value  
if TERMINAL_TEST(state) then return UTILITY(state)  
 $v \leftarrow \infty$   
for each  $a$  in ACTIONS(state) do  
     $v \leftarrow \text{MIN}(v, \text{MAX\_VALUE}(\text{RESULTS}(state, a)))$   
return  $v$ 
```