

# FAI LAB 5

## Adversarial search

Paolo Morettin

2024-25

**function** MINIMAX-SEARCH(*game, state*) **returns** *an action*

*player*  $\leftarrow$  *game*.TO-MOVE(*state*)

*value, move*  $\leftarrow$  MAX-VALUE(*game, state*)

**return** *move*

**function** MAX-VALUE(*game, state*) **returns** *a (utility, move) pair*

**if** *game*.IS-TERMINAL(*state*) **then return** *game*.UTILITY(*state, player*), *null*

*v*  $\leftarrow -\infty$

**for each** *a* **in** *game*.ACTIONS(*state*) **do**

*v2, a2*  $\leftarrow$  MIN-VALUE(*game, game*.RESULT(*state, a*))

**if** *v2* > *v* **then**

*v, move*  $\leftarrow$  *v2, a*

**return** *v, move*

**function** MIN-VALUE(*game, state*) **returns** *a (utility, move) pair*

**if** *game*.IS-TERMINAL(*state*) **then return** *game*.UTILITY(*state, player*), *null*

*v*  $\leftarrow +\infty$

**for each** *a* **in** *game*.ACTIONS(*state*) **do**

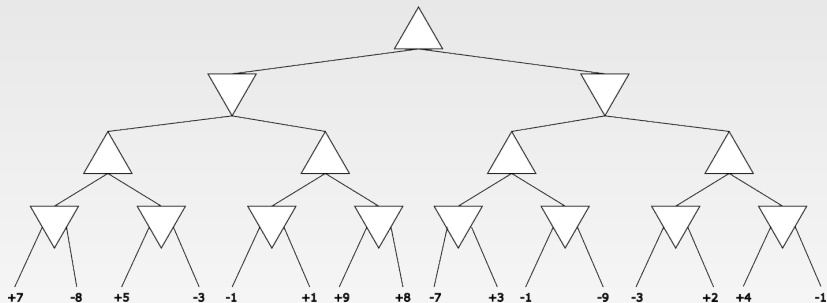
*v2, a2*  $\leftarrow$  MAX-VALUE(*game, game*.RESULT(*state, a*))

**if** *v2* < *v* **then**

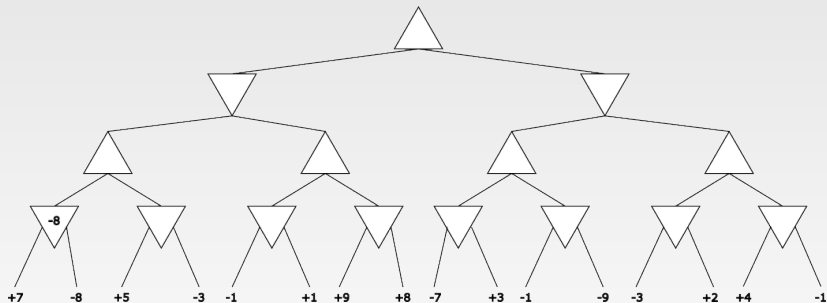
*v, move*  $\leftarrow$  *v2, a*

**return** *v, move*

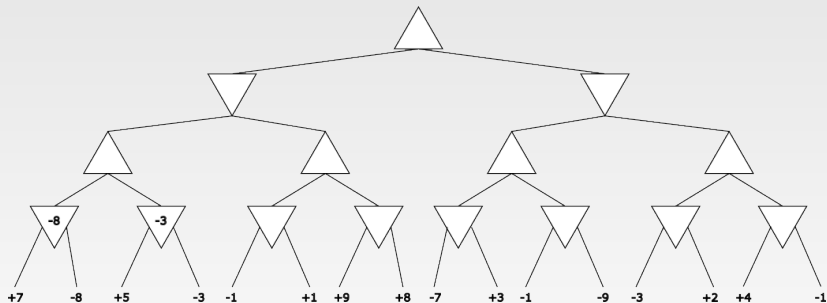
# Minimax



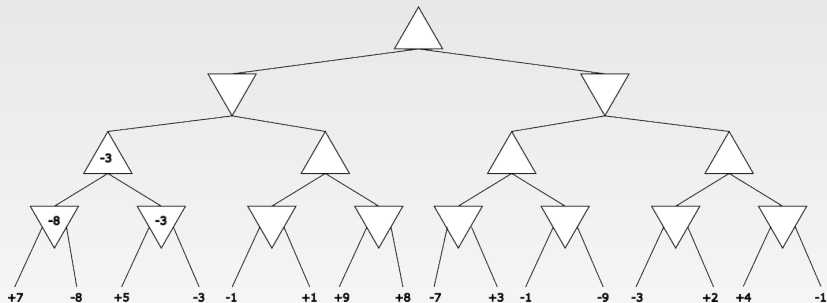
# Minimax



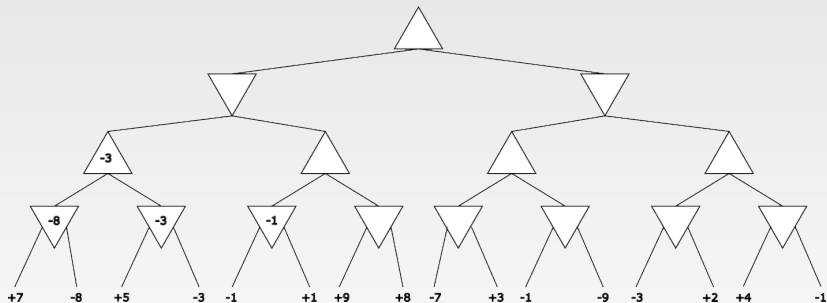
# Minimax



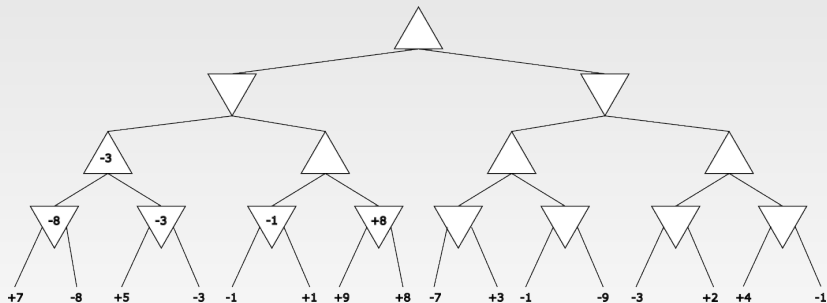
# Minimax



# Minimax

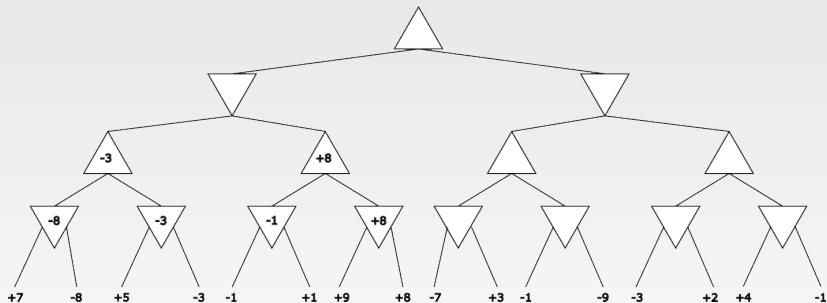


# Minimax

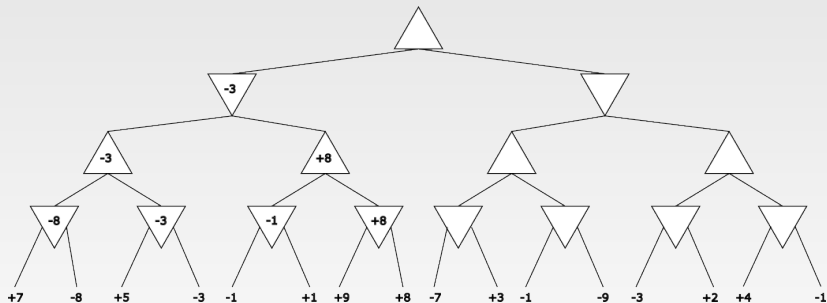




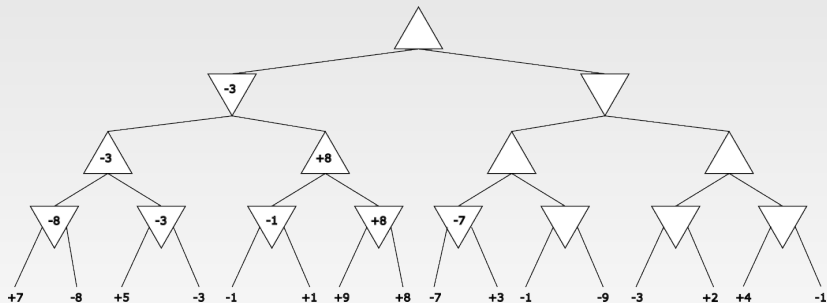
# Minimax



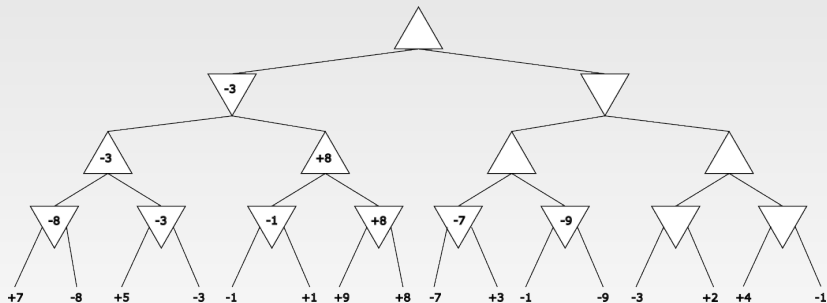
# Minimax



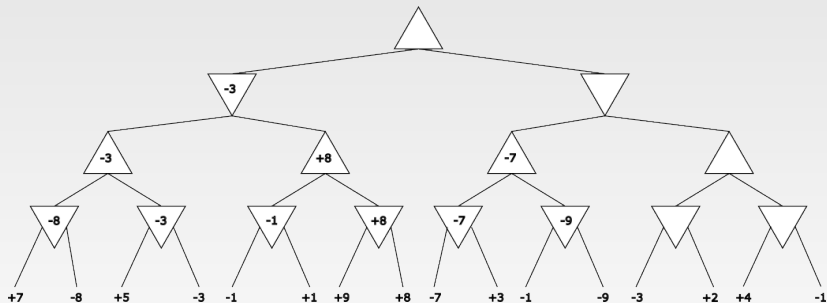
# Minimax



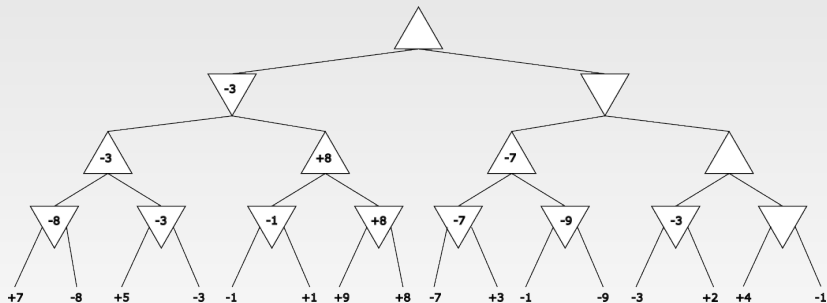
# Minimax



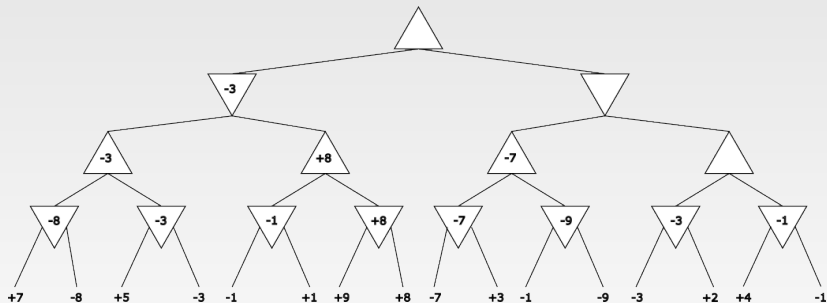
# Minimax



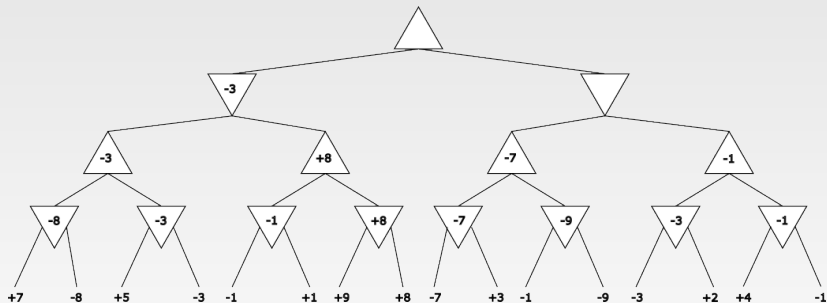
# Minimax



# Minimax

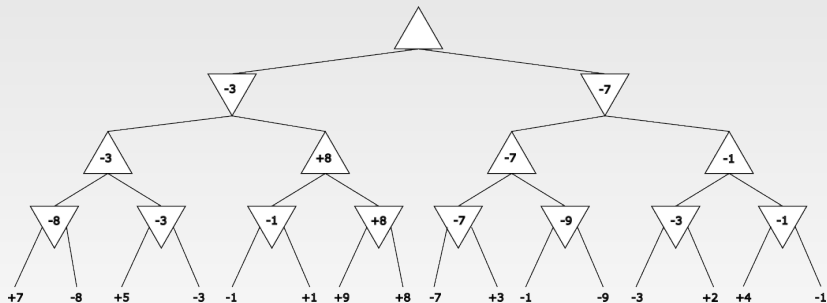


# Minimax

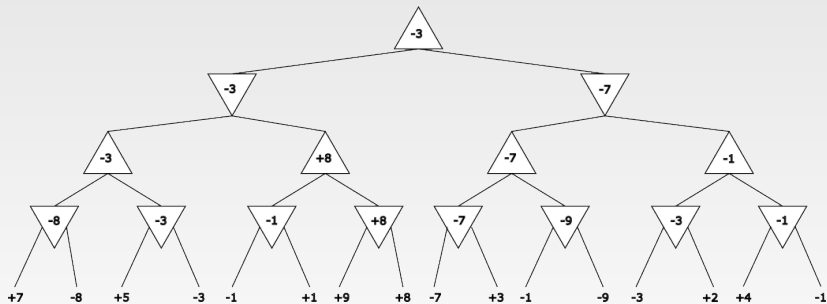




# Minimax



# Minimax



# $\alpha - \beta$ pruning

**function** ALPHA-BETA-SEARCH(*game*, *state*) **returns** an action

$\text{player} \leftarrow \text{game}.\text{TO-MOVE}(\text{state})$

$\text{value}, \text{move} \leftarrow \text{MAX-VALUE}(\text{game}, \text{state}, -\infty, +\infty)$

**return** *move*

**function** MAX-VALUE(*game*, *state*,  $\alpha$ ,  $\beta$ ) **returns** a (*utility*, *move*) pair

**if** *game*.IS-TERMINAL(*state*) **then return** *game*.UTILITY(*state*, *player*), null

$v \leftarrow -\infty$

**for each** *a* **in** *game*.ACTIONS(*state*) **do**

$v2, a2 \leftarrow \text{MIN-VALUE}(\text{game}, \text{game}.\text{RESULT}(\text{state}, a), \alpha, \beta)$

**if**  $v2 > v$  **then**

$v, \text{move} \leftarrow v2, a$

$\alpha \leftarrow \text{MAX}(\alpha, v)$

**if**  $v \geq \beta$  **then return** *v*, *move*

**return** *v*, *move*

**function** MIN-VALUE(*game*, *state*,  $\alpha$ ,  $\beta$ ) **returns** a (*utility*, *move*) pair

**if** *game*.IS-TERMINAL(*state*) **then return** *game*.UTILITY(*state*, *player*), null

$v \leftarrow +\infty$

**for each** *a* **in** *game*.ACTIONS(*state*) **do**

$v2, a2 \leftarrow \text{MAX-VALUE}(\text{game}, \text{game}.\text{RESULT}(\text{state}, a), \alpha, \beta)$

**if**  $v2 < v$  **then**

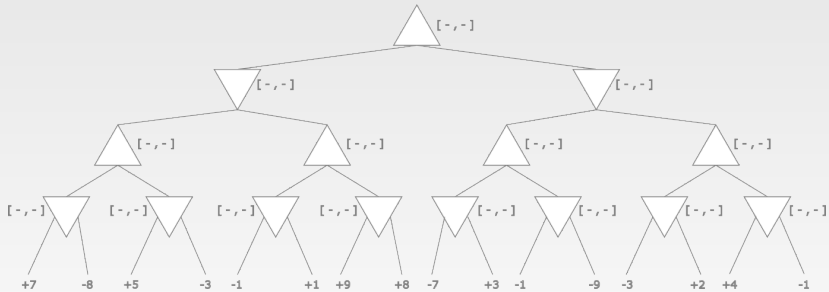
$v, \text{move} \leftarrow v2, a$

$\beta \leftarrow \text{MIN}(\beta, v)$

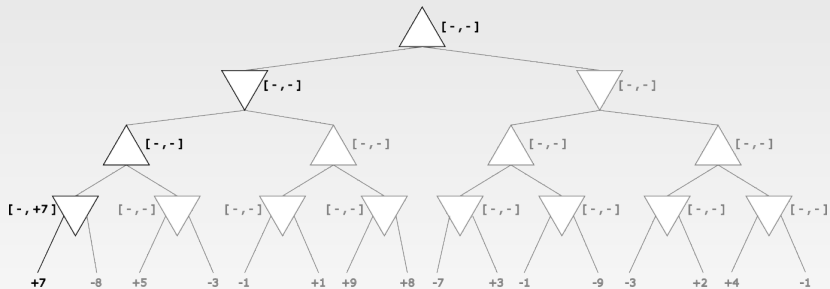
**if**  $v \leq \alpha$  **then return** *v*, *move*

**return** *v*, *move*

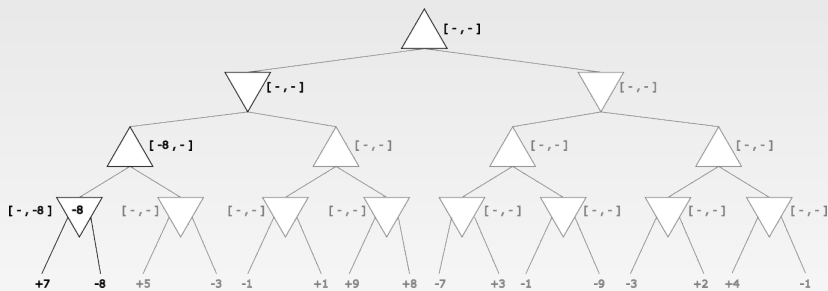
## $\alpha - \beta$ pruning



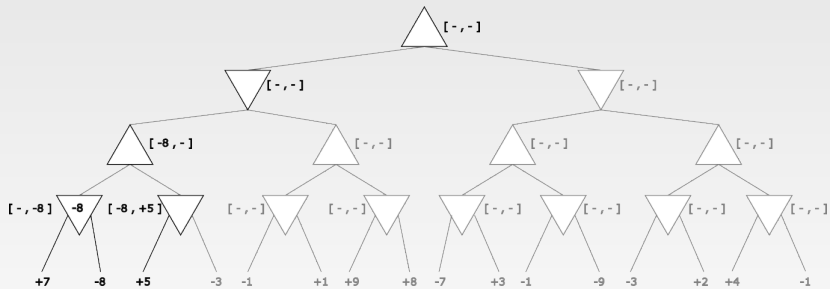
# $\alpha - \beta$ pruning



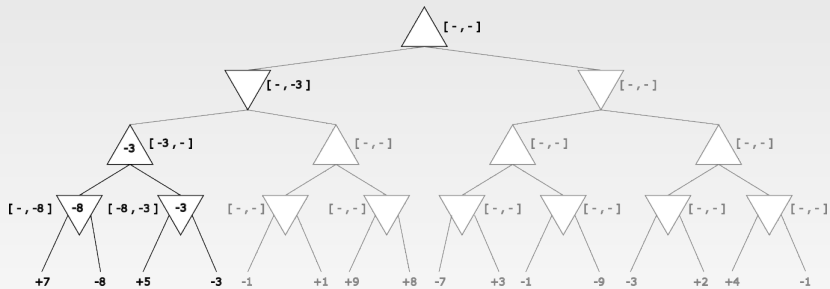
# $\alpha - \beta$ pruning



# $\alpha - \beta$ pruning

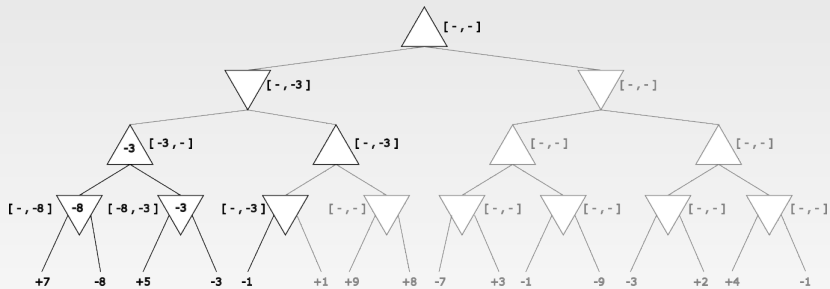


# $\alpha - \beta$ pruning

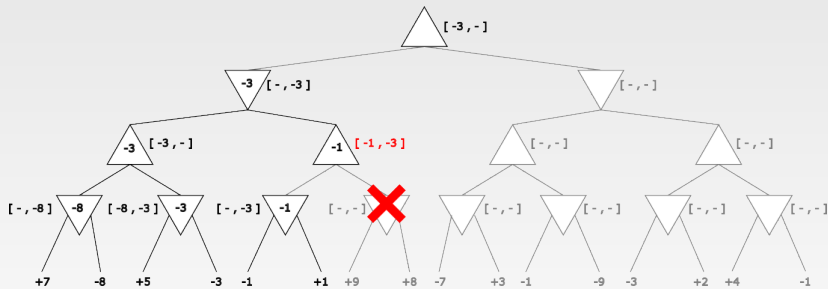




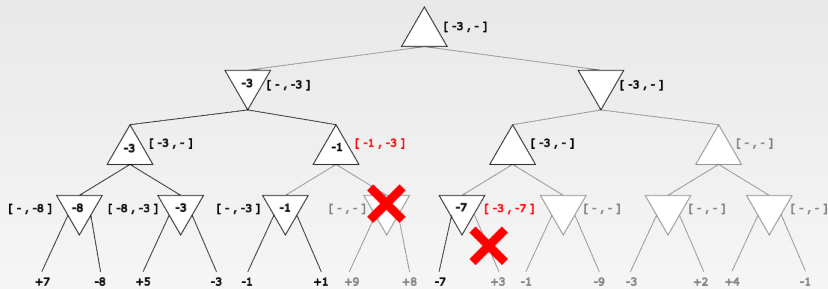
# $\alpha - \beta$ pruning



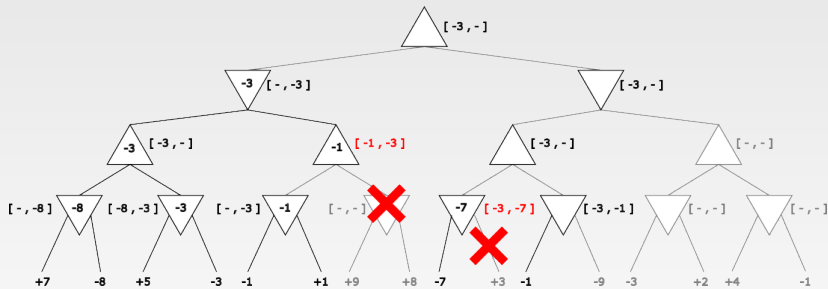
# $\alpha - \beta$ pruning



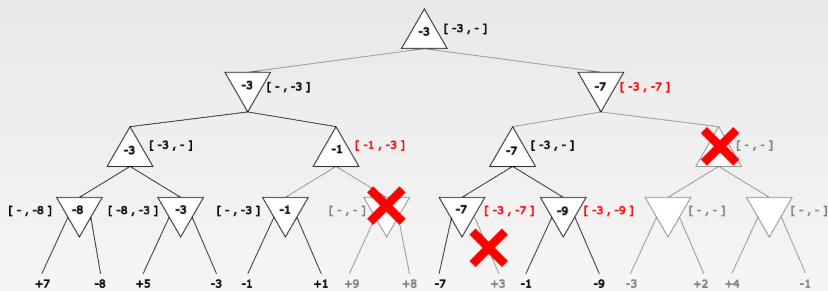
# $\alpha - \beta$ pruning



# $\alpha - \beta$ pruning



# $\alpha - \beta$ pruning



# Simple dice game

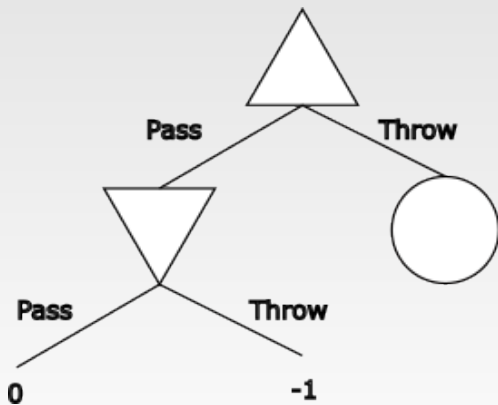
- both *MAX* and *MIN* players have 2 moves:
  - *throw* an even d6 dice
  - *pass*
- the game is draw (0) if both passed or the two throws are the same
- otherwise the player with highest throw wins (either +1 or -1)
- *MAX goes first*

# Simple dice game

Should *MAX* pass?

# Simple dice game

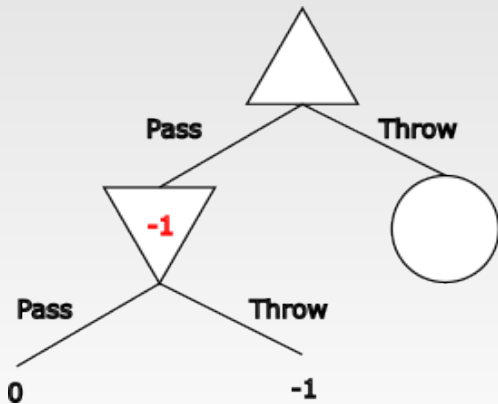
Should *MAX* pass?





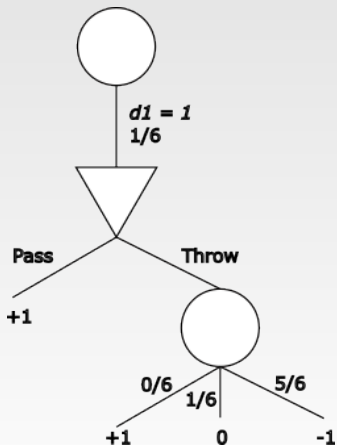
# Simple dice game

Should *MAX* pass?



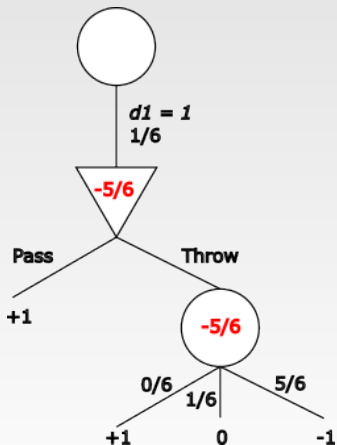
# Simple dice game

Should *MAX* throw the d6?



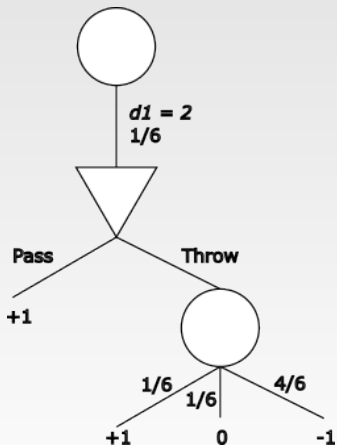
# Simple dice game

Should *MAX* throw the d6?



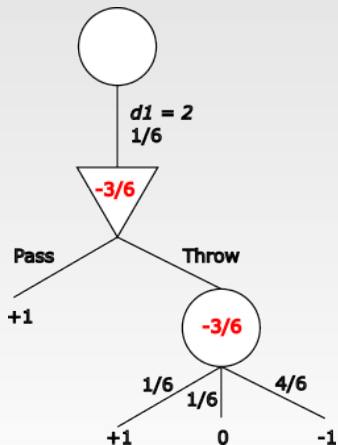
# Simple dice game

Should *MAX* throw the d6?



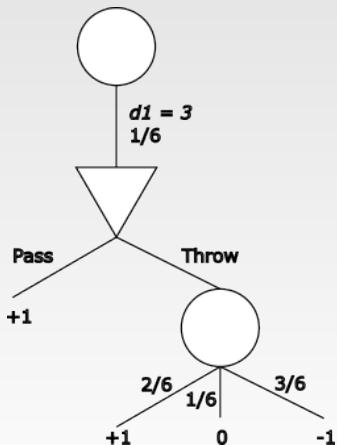
# Simple dice game

Should *MAX* throw the d6?



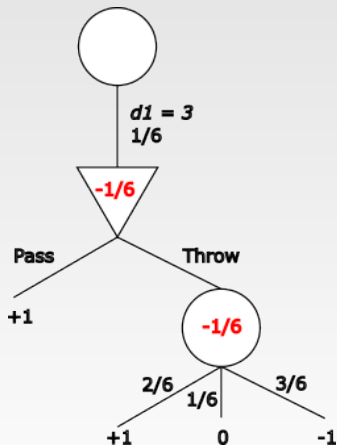
# Simple dice game

Should *MAX* throw the d6?



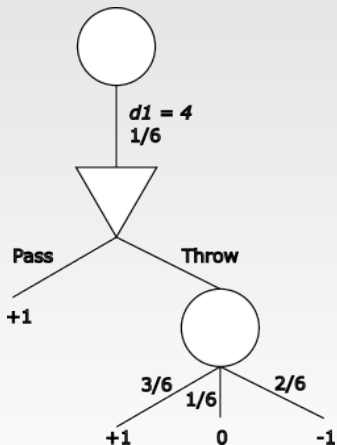
# Simple dice game

Should *MAX* throw the d6?



# Simple dice game

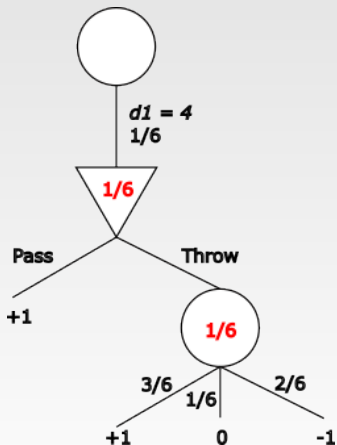
Should *MAX* throw the d6?





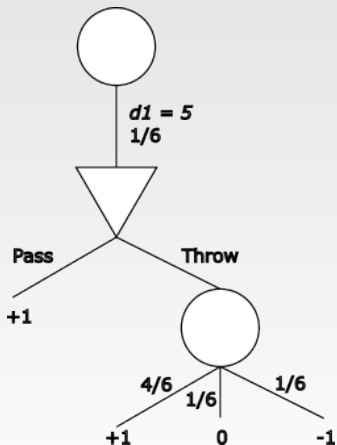
# Simple dice game

Should *MAX* throw the d6?



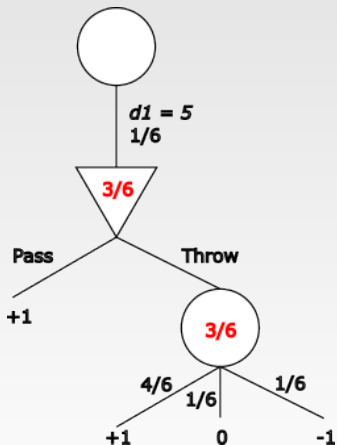
# Simple dice game

Should *MAX* throw the d6?



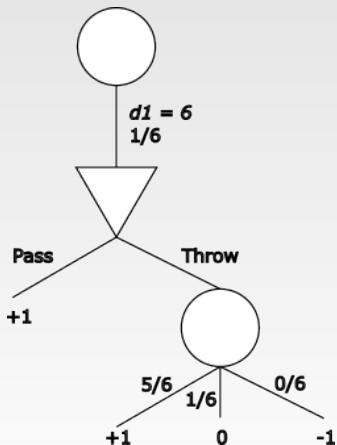
# Simple dice game

Should *MAX* throw the d6?



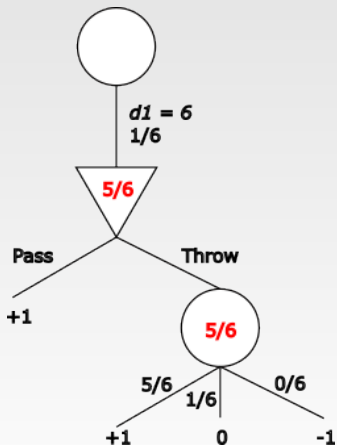
# Simple dice game

Should *MAX* throw the d6?



# Simple dice game 1.0

Should *MAX* throw the d6?



# Simple dice game 1.0

Should *MAX* throw the d6? Always!

