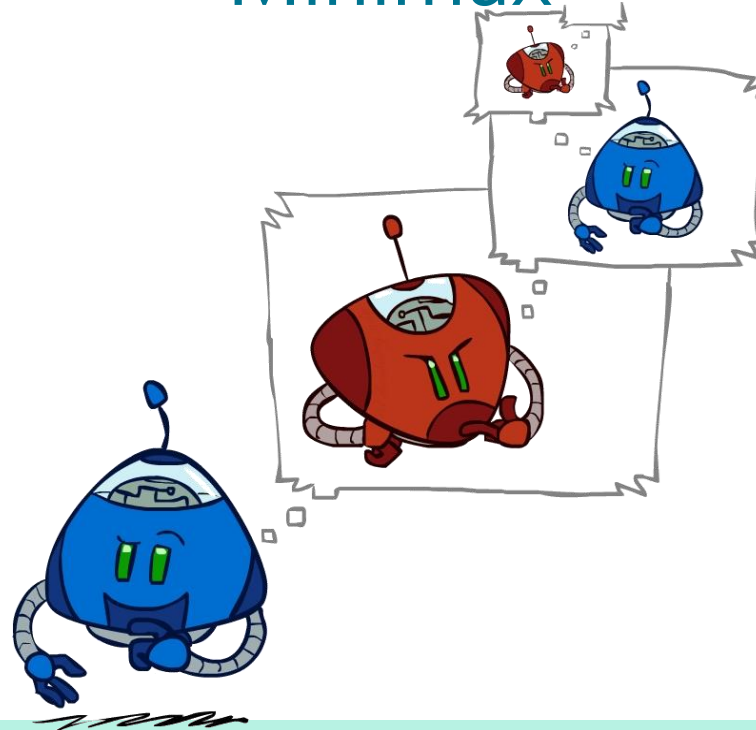


Advanced Topics in AI

Minimax

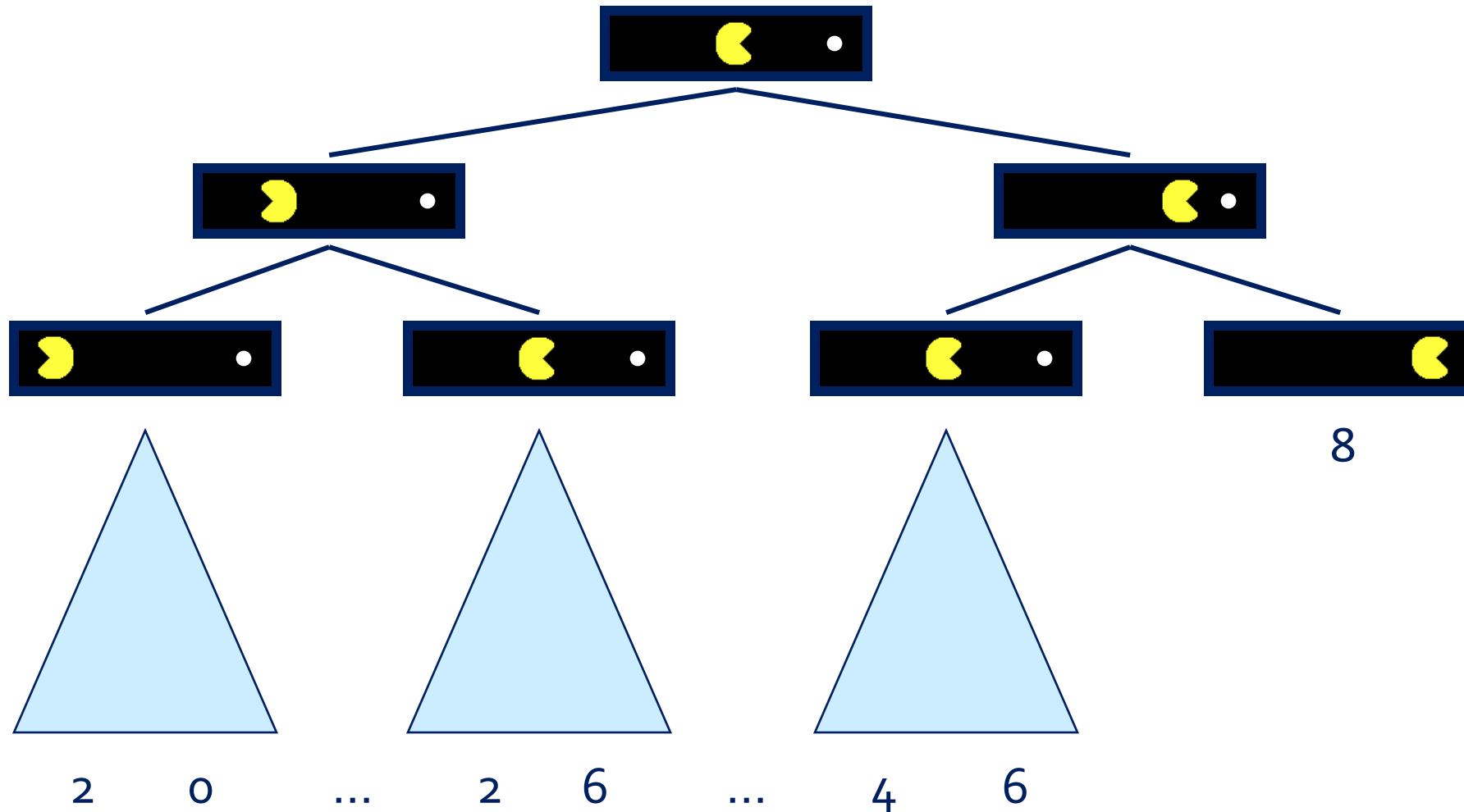


Instructor: Prof. Dr. techn. Wolfgang Nejdl
Leibniz University Hannover



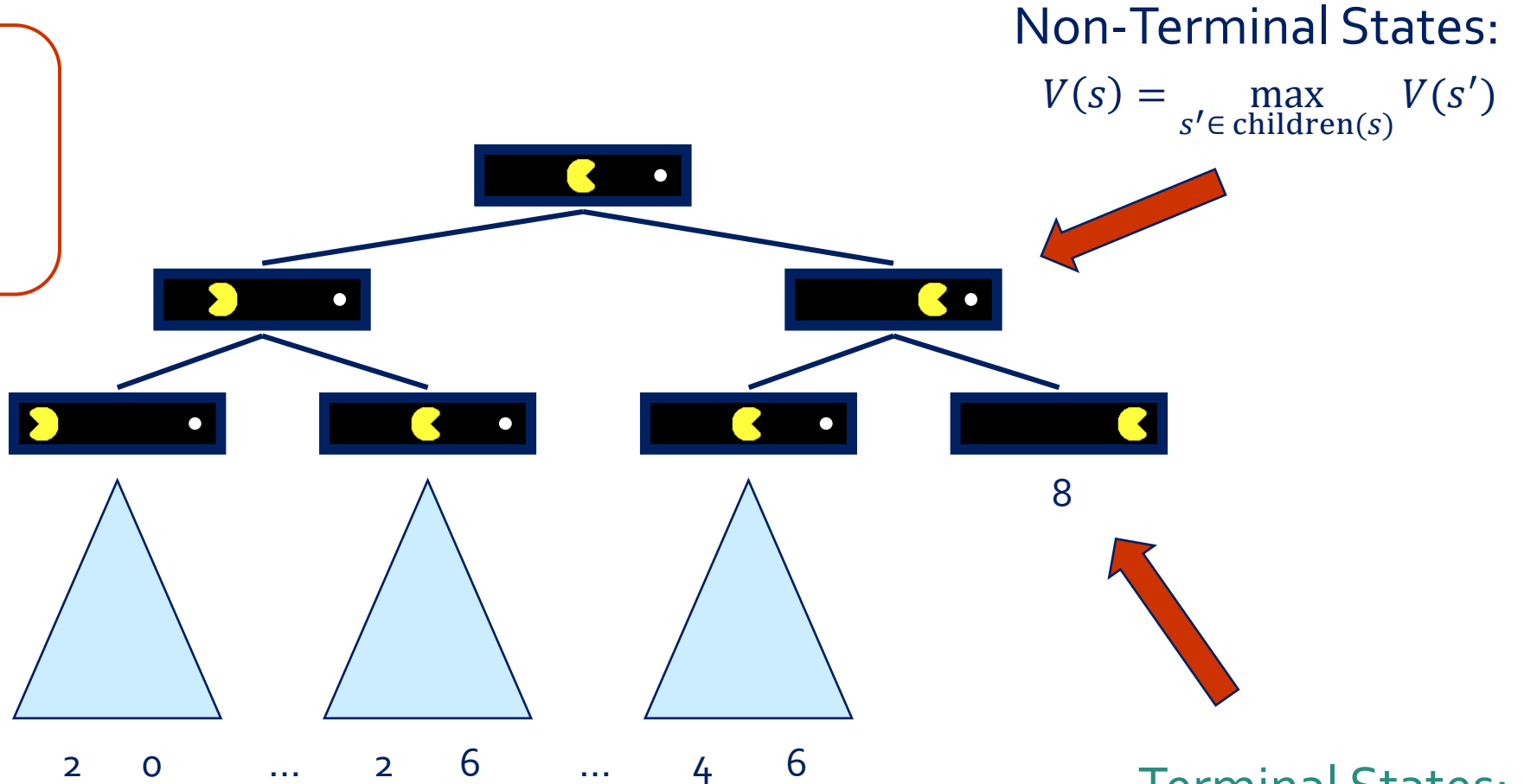
[These slides were created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley. All materials are available at <http://ai.berkeley.edu>.]

Single-Agent Trees

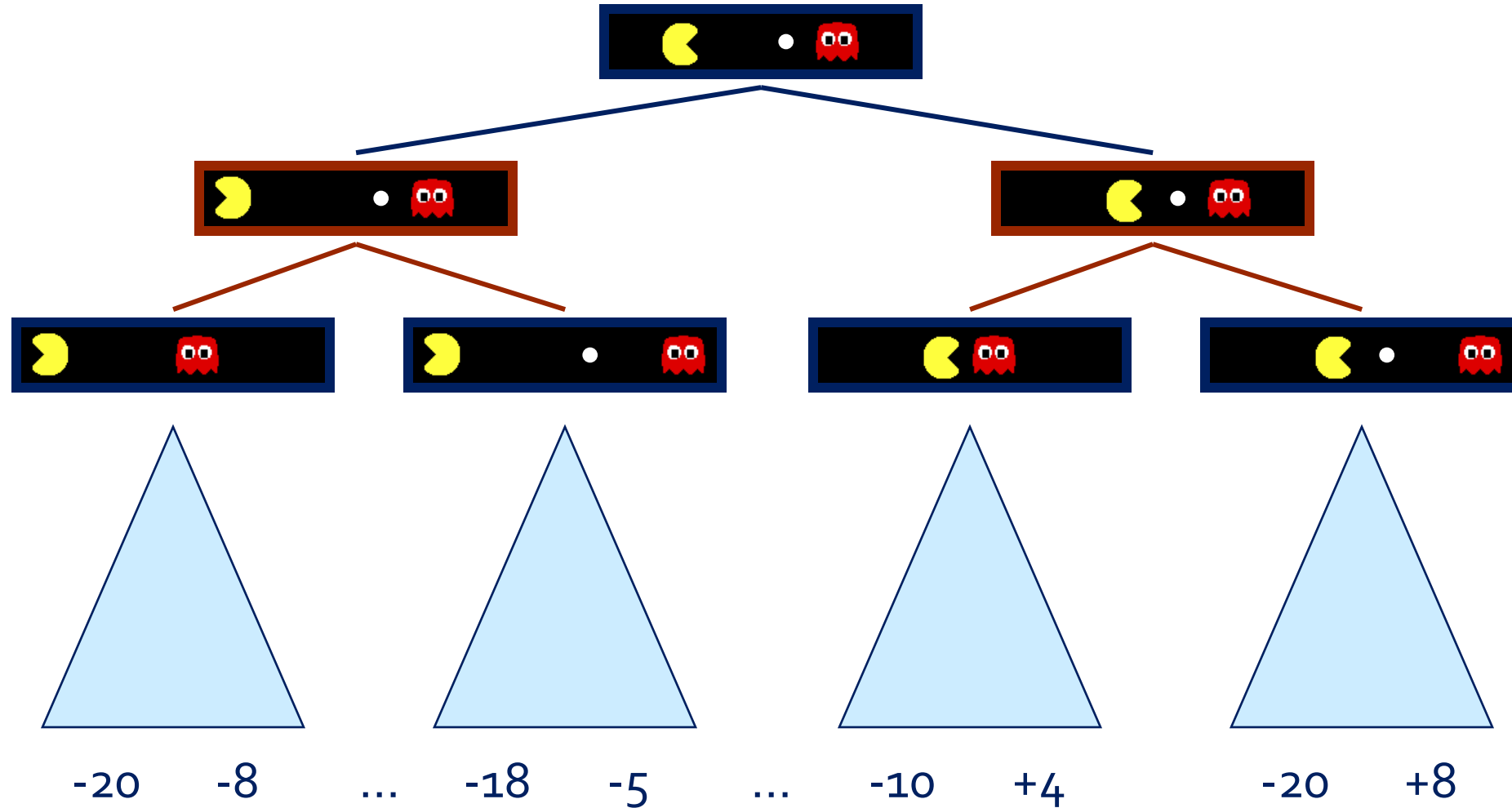


Value of a State

Value of a state:
The best achievable outcome (utility) from that state



Adversarial Game Trees



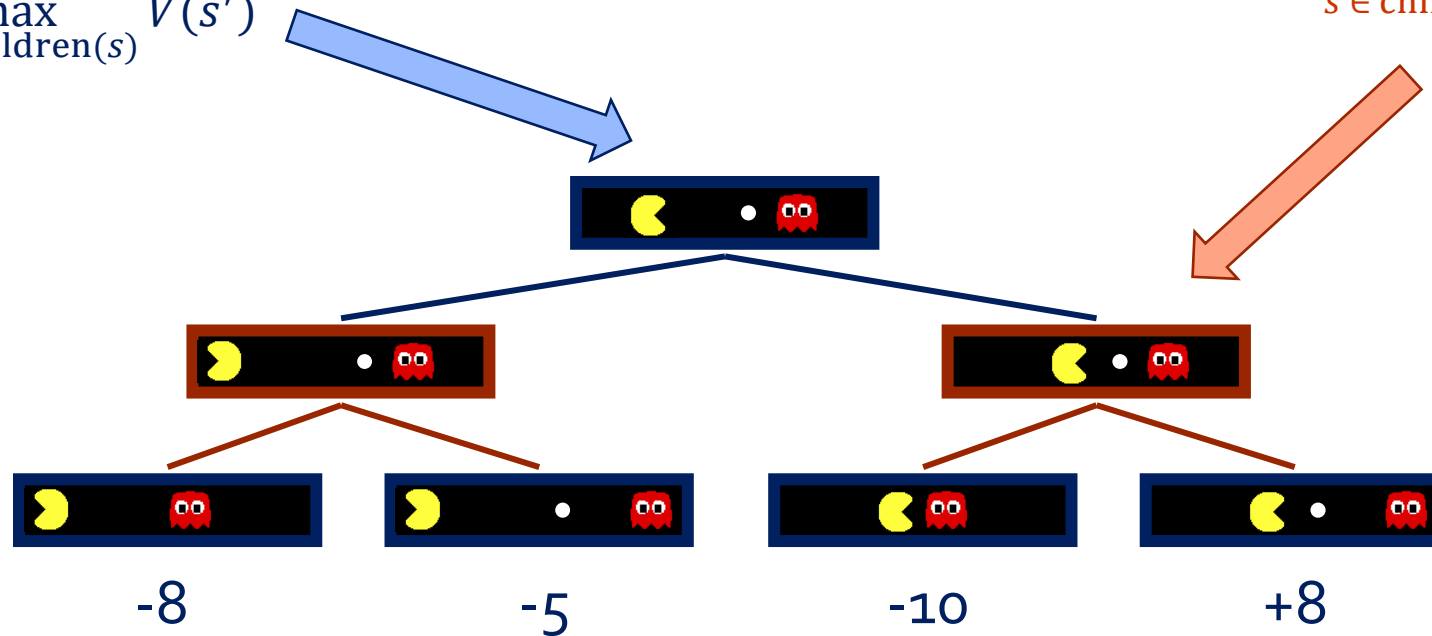
Minimax Values

States under Agent's Control:

$$V(s) = \max_{s' \in \text{children}(s)} V(s')$$

States Under Opponent's Control:

$$V(s') = \min_{s \in \text{children}(s')} V(s)$$



Terminal States:

$$V(s) = \text{known}$$

Tic-Tac-Toe Game Tree



MAX (X)



MIN (O)



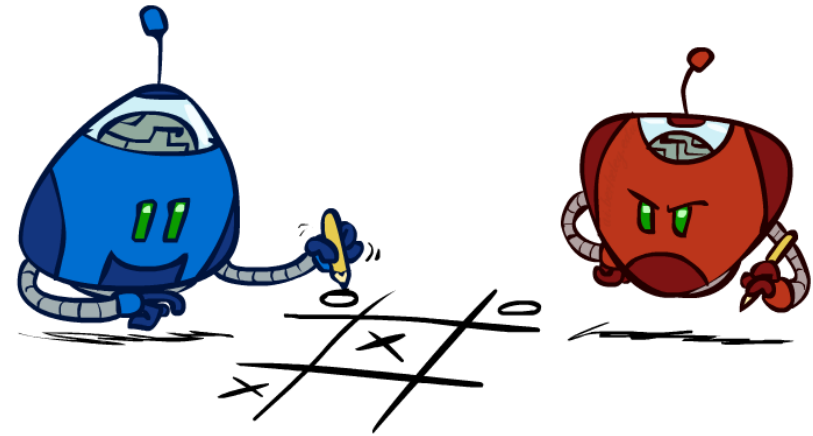
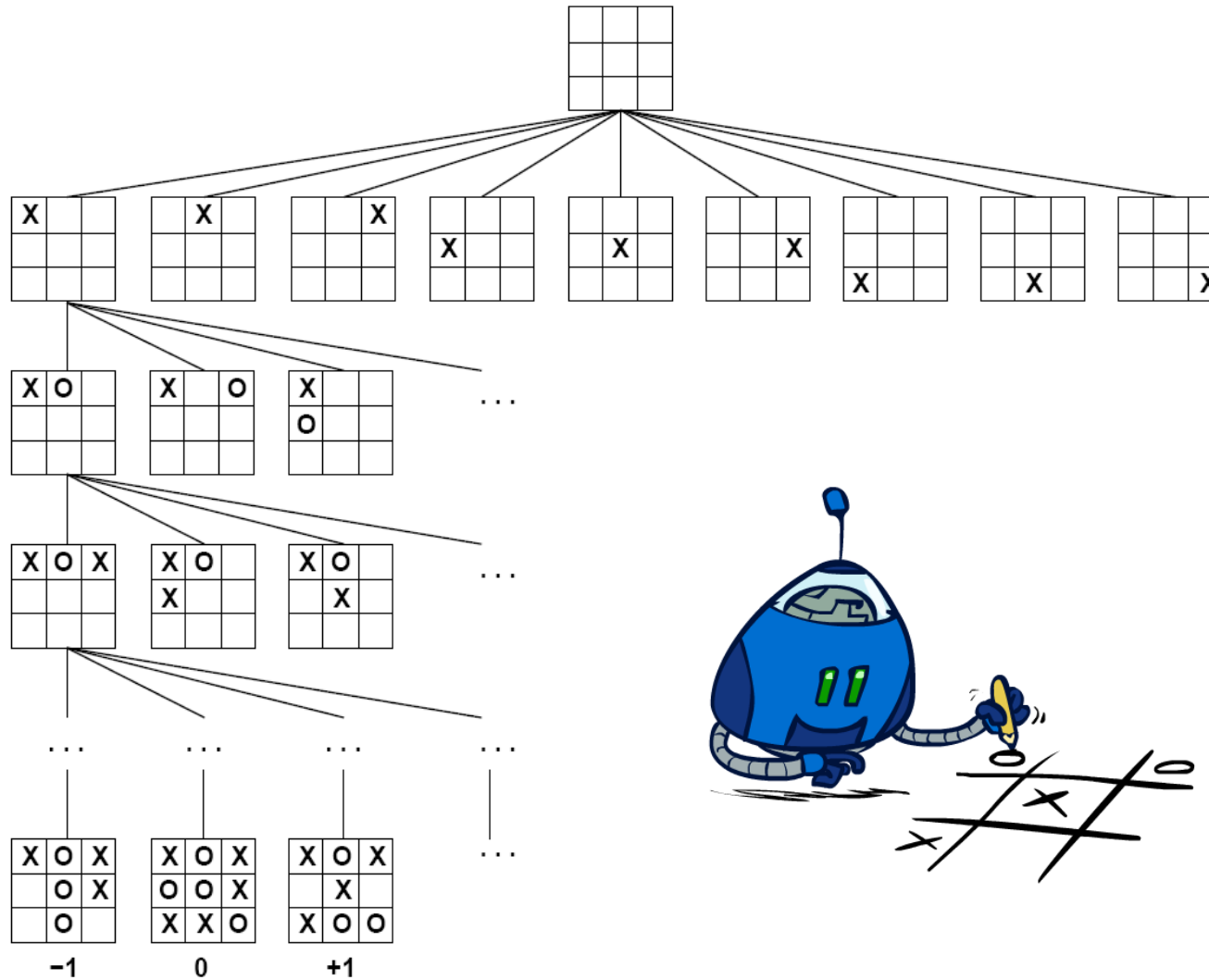
MAX (X)



MIN (O)

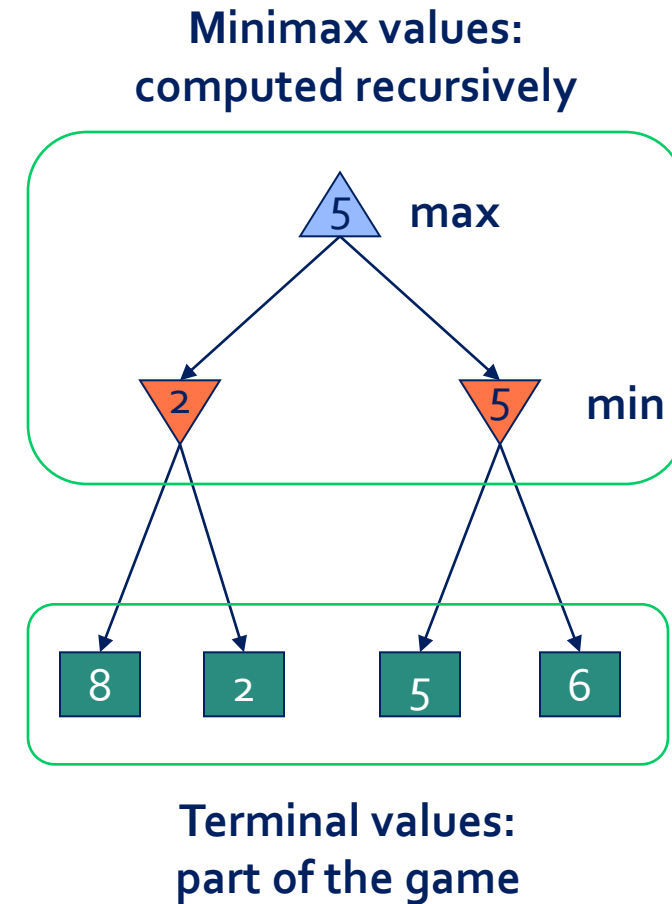
TERMINAL

Utility



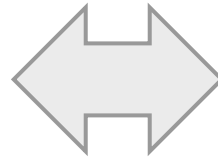
Adversarial Search (Minimax)

- Deterministic, zero-sum games:
 - Tic-tac-toe, chess, checkers
 - One player maximizes result
 - The other minimizes result
- Minimax search:
 - A state-space search tree
 - Players alternate turns
 - Compute each node's **minimax value**: the best achievable utility against a rational (optimal) adversary



Minimax Implementation

```
def max-value(state):  
    initialize v = -∞  
    for each successor of state:  
        v = max(v, min-value(successor))  
    return v
```



```
def min-value(state):  
    initialize v = +∞  
    for each successor of state:  
        v = min(v, max-value(successor))  
    return v
```

$$V(s) = \max_{s' \in \text{children}(s)} V(s')$$

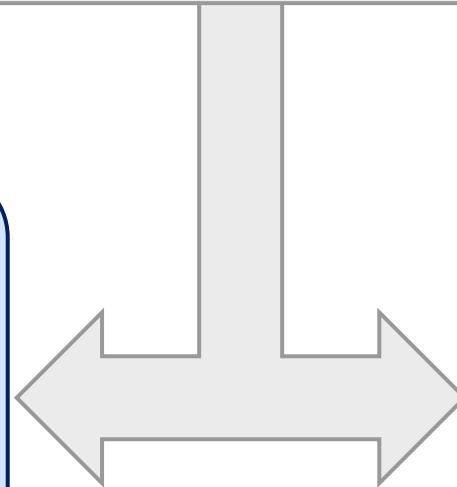
$$V(s') = \min_{s \in \text{children}(s')} V(s)$$

Minimax Implementation (Dispatch)

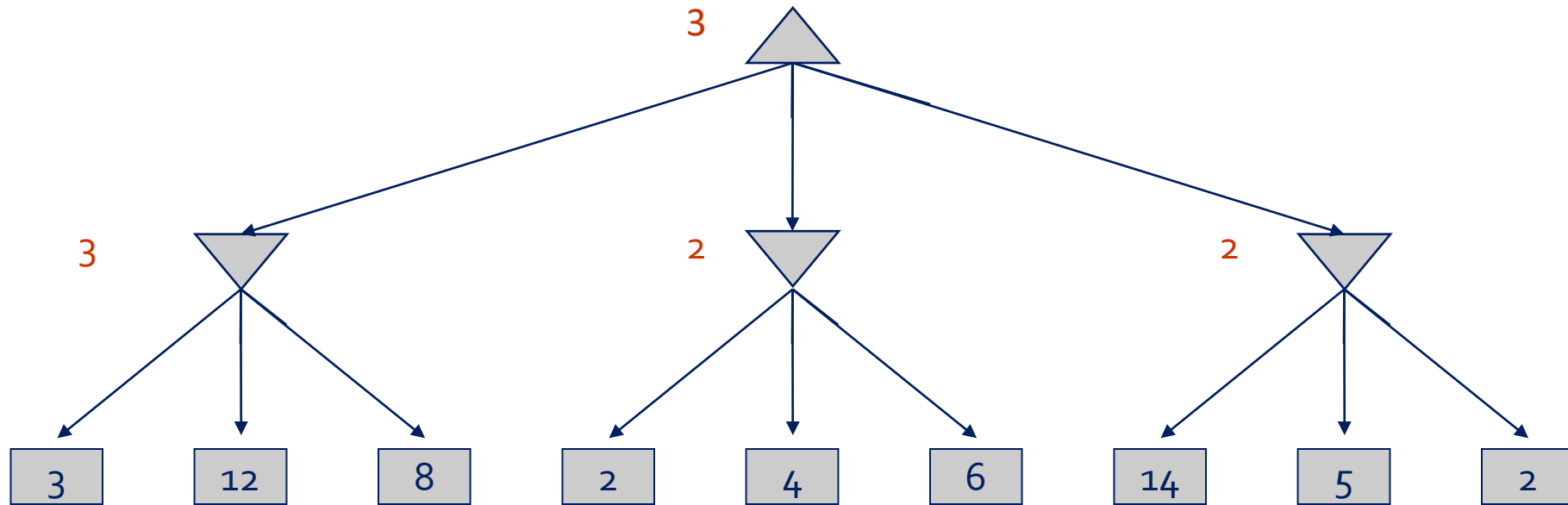
```
def value(state):  
    if the state is a terminal state: return the state's utility  
    if the next agent is MAX: return max-value(state)  
    if the next agent is MIN: return min-value(state)
```

```
def max-value(state):  
    initialize  $v = -\infty$   
    for each successor of state:  
         $v = \max(v, \text{value}(\text{successor}))$   
    return  $v$ 
```

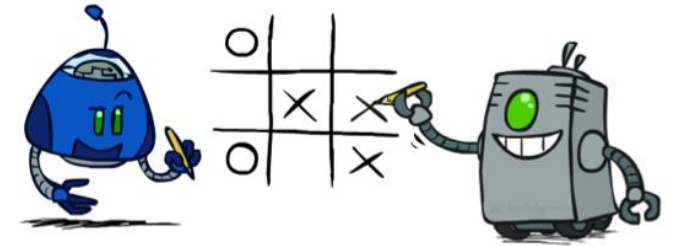
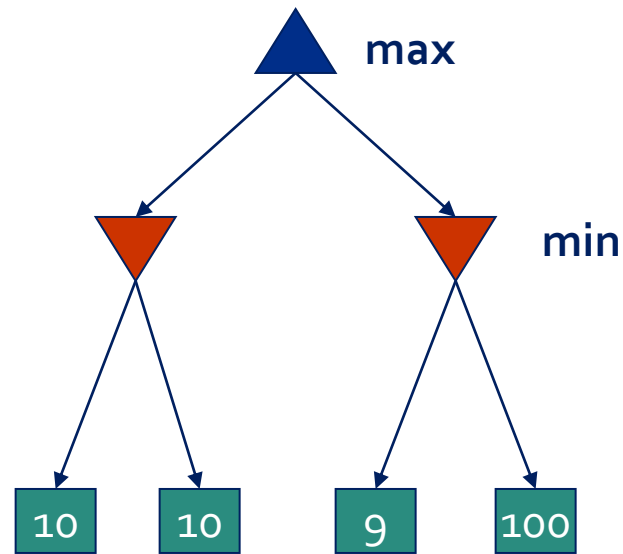
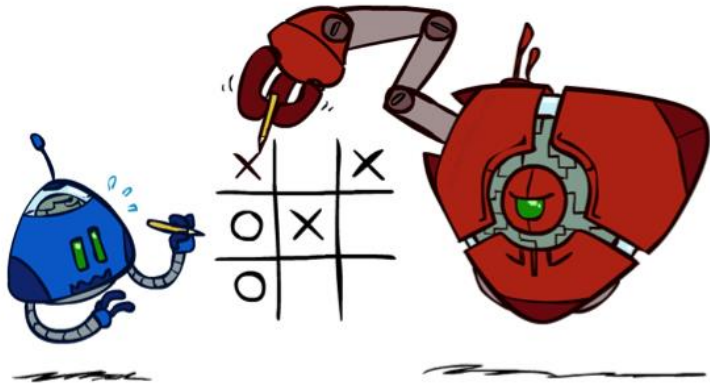
```
def min-value(state):  
    initialize  $v = +\infty$   
    for each successor of state:  
         $v = \min(v, \text{value}(\text{successor}))$   
    return  $v$ 
```



Minimax Example



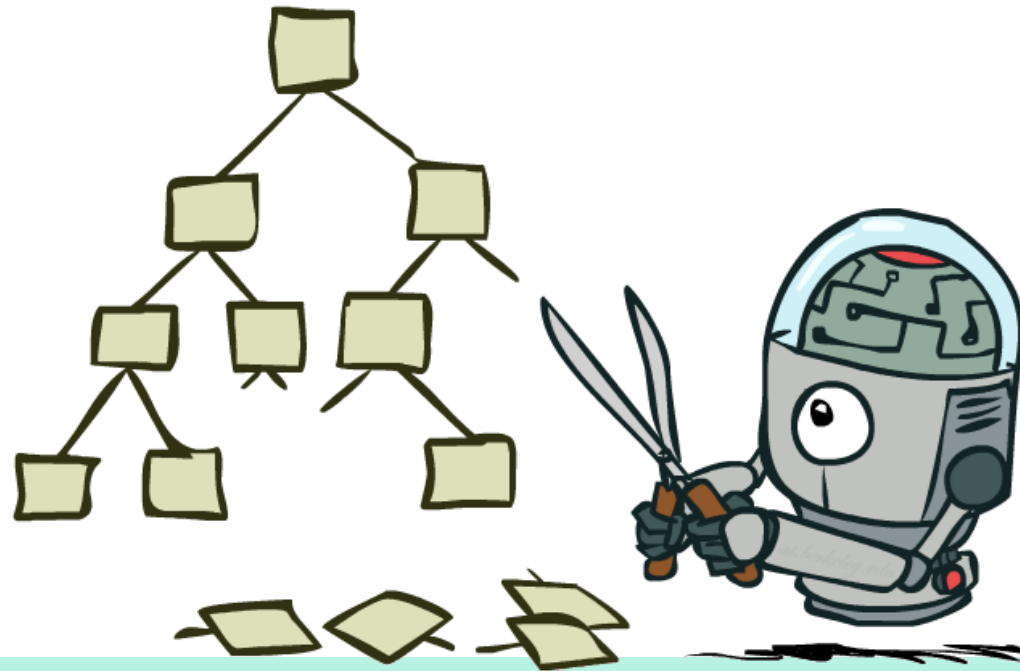
Minimax Properties



Optimal against a perfect player. Otherwise?

Advanced Topics in AI

Next: Alpha-Beta Pruning



Instructor: Prof. Dr. techn. Wolfgang Nejdl

Leibniz University Hannover



[These slides were created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley. All materials are available at <http://ai.berkeley.edu>.]