

Class ModeloJuego DetTZ2:

```
def Inicializa (self)
    return
```

```
def acciones_legales (self, s, j)
    return { a1 ... an }
```

```
def transición (Selt, s, a, i)
    return s'
```

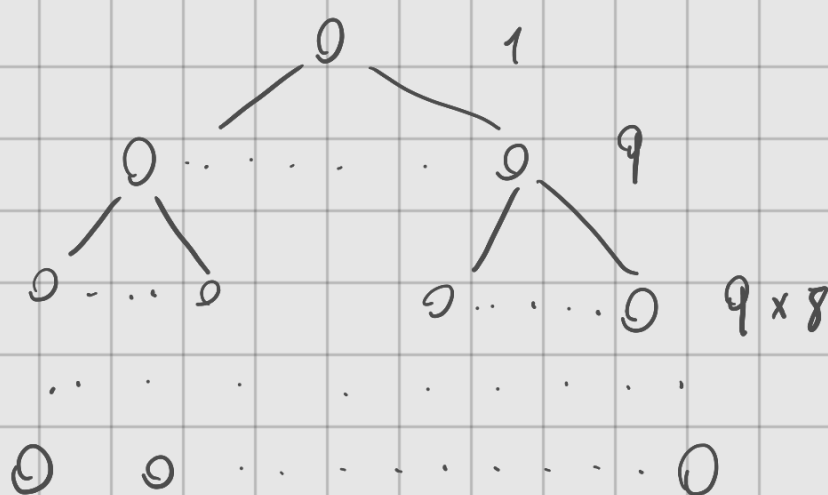
```
def terminal(self, s):  
    return bool
```

```
def ganancia (self, s)
    return ganancia de 1
```

Juego gato

$$x = (0, 0, 0, 0, 0, 0, 0, 0, 0)$$

X	0
X	

$$\Rightarrow x = (1, -1, 0, 1, 0, \dots, 0)$$


En profundidad tenemos

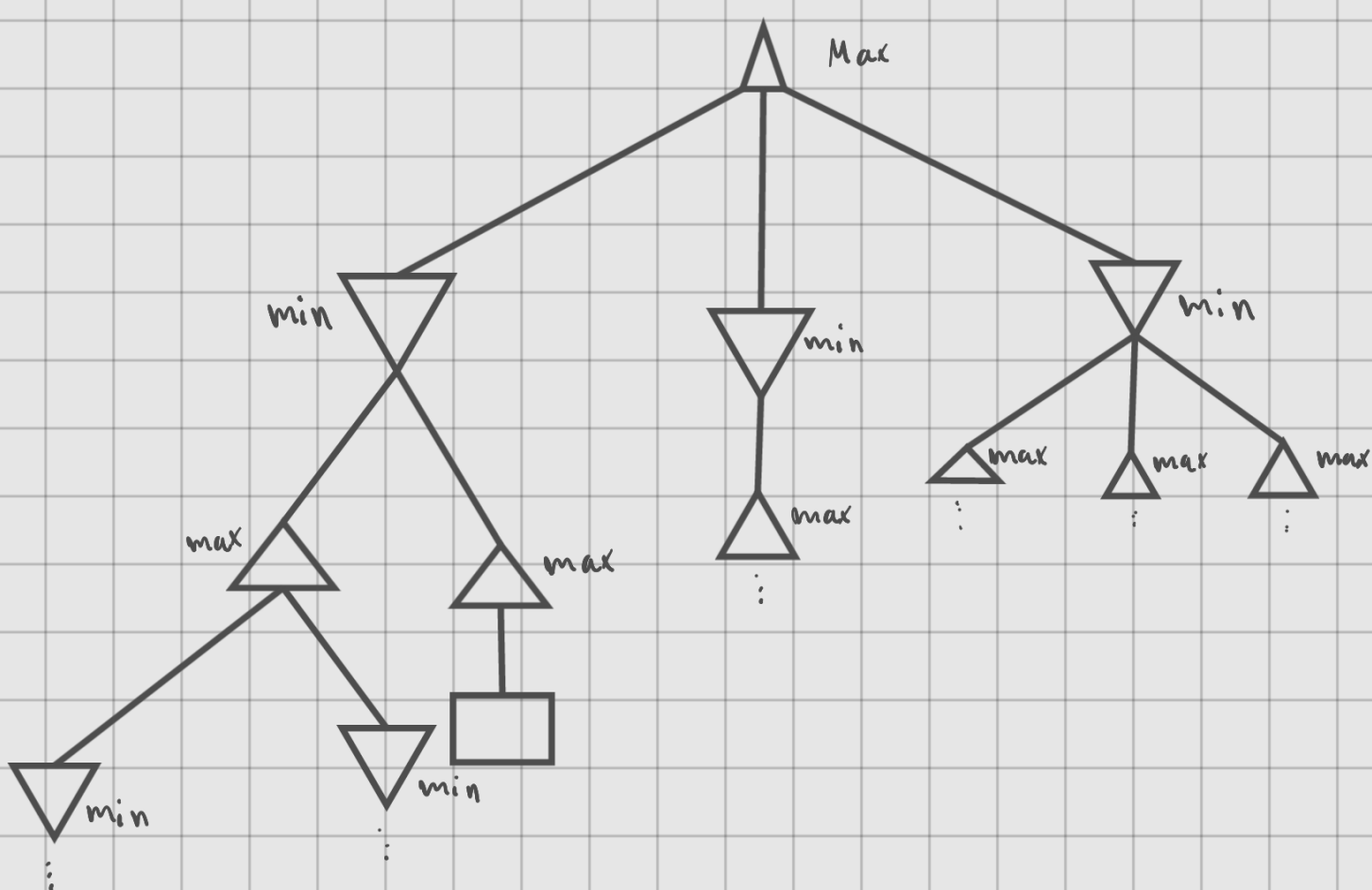
algo menor a  $q!$

Hay estados terminales en algunas ramas, por eso no llega a 9!

Minimizar la ganancia del oponente = Maximizar mi ganancia

## Adversal Search (Minimax)

Asumimos que el rival Sabe jugar, por lo que el también va a jugar queriendo maximizar su ganancia



Es óptimo contra un jugador perfecto

No toma riesgos

## Complejidad

Tiempo  $O(b^m)$

Espacio  $O(bm)$