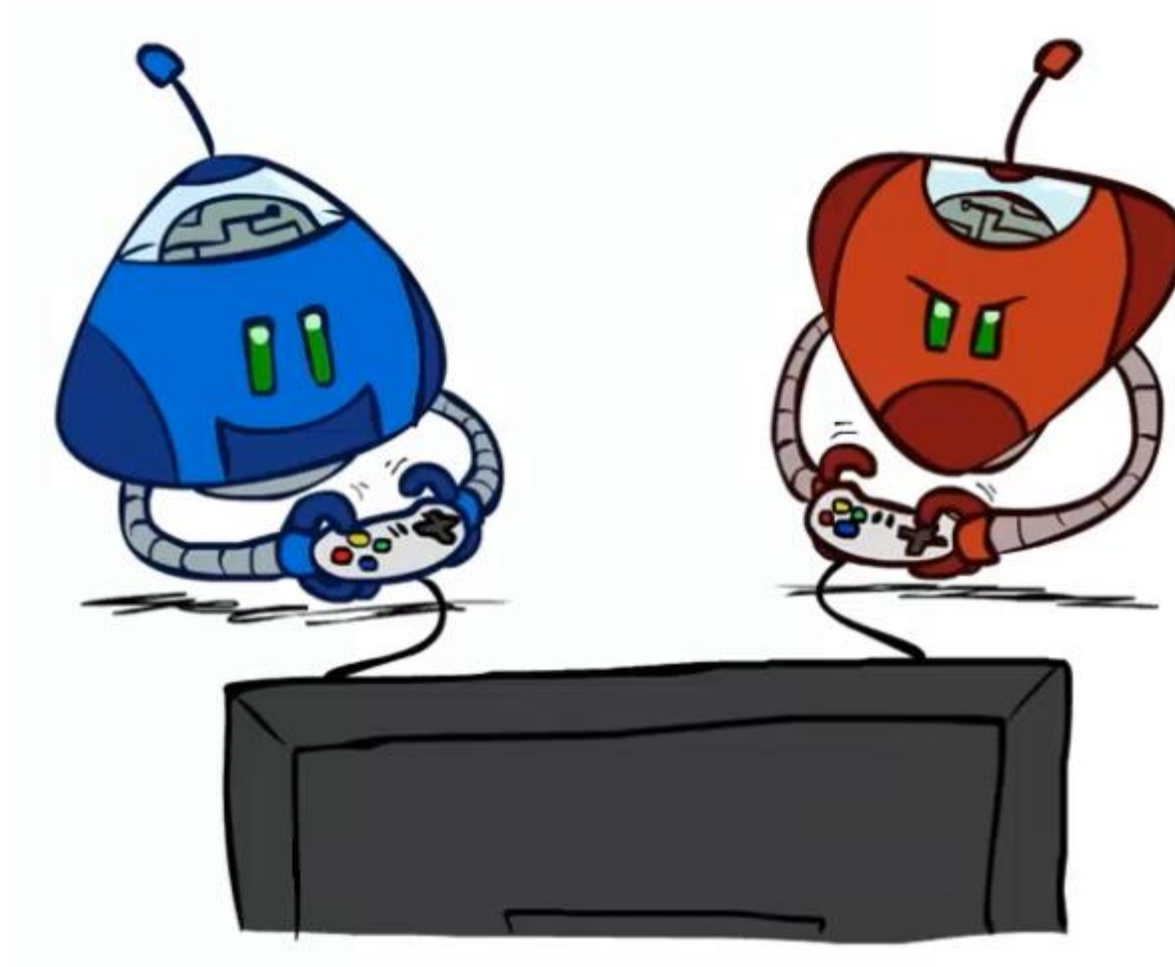


بازی‌ها و جستجوی رقابتی

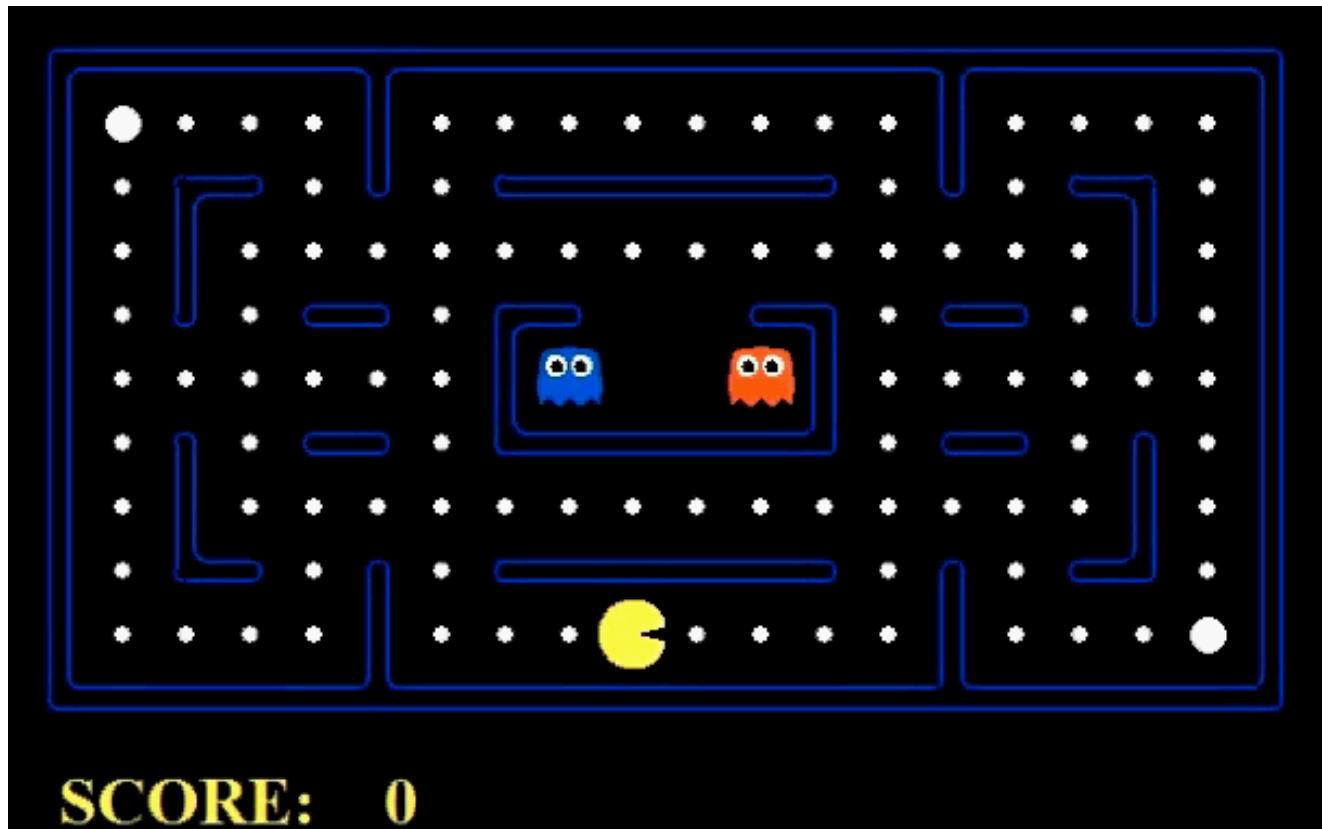
سید ناصر رضوی n.razavi@tabrizu.ac.ir

۱۳۹۵



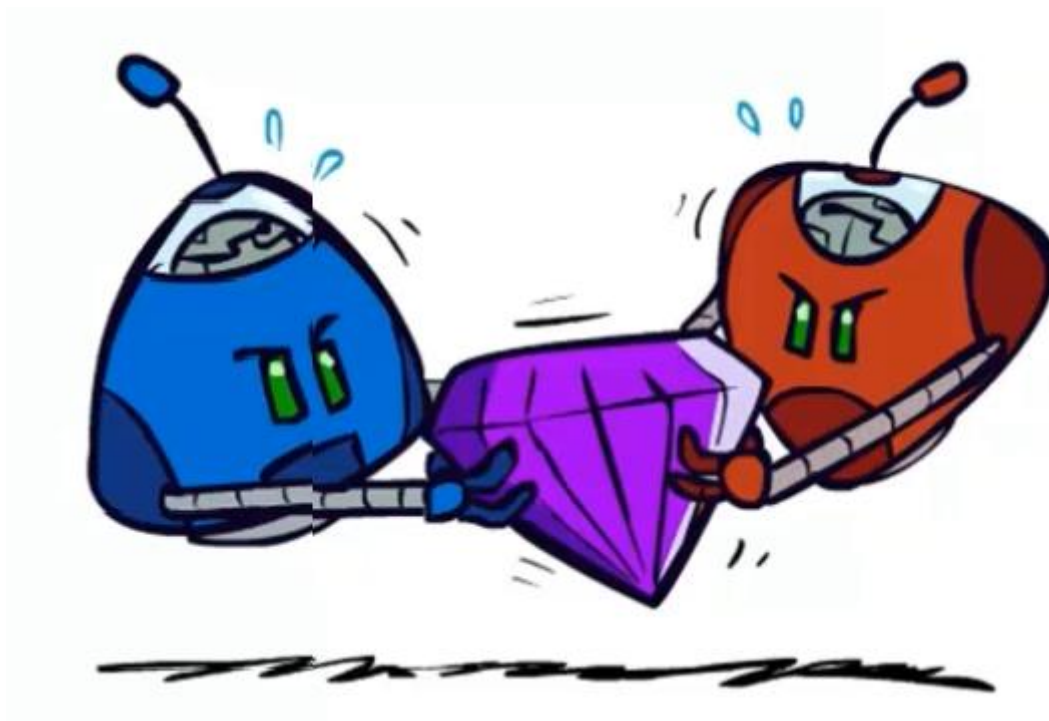
محاسبه رفتار

۳



بازی‌های رقابتی

۴



انواع بازی‌ها

۵

□ انواع بسیار مختلفی از بازی‌ها وجود دارد!

□ ویژگی‌ها.

□ قطعی یا اتفاقی؟

□ یک نفره، دو نفره، یا چند نفره

□ مجموع صفر؟

□ اطلاعات کامل (حالت محیط قابل مشاهده)



□ هدف. یافتن الگوریتمی که بتواند در هر حالت از بازی یک **استراتژی** (بهینه) پیشنهاد کند.

بازی‌های قطعی

۶



□ بیان بازی به شکل رسمی.

□ حالت‌ها: S

□ بازیکن‌ها: $P = \{1, \dots, N\}$

□ اعمال: A

□ تابع تغییر حالت: $S \times A \rightarrow S$

□ تابع تست حالت‌های پایانی: $S \rightarrow \{T, F\}$

□ تابع سودمندی حالت‌های پایانی: $S \times P \rightarrow R$

□ راه‌حل برای یک بازیکن.

□ یک **سیاست** به صورت تابعی از حالت‌ها به اعمال.

Policy: $S \rightarrow A$

بازی‌های مجموع صفر

۷



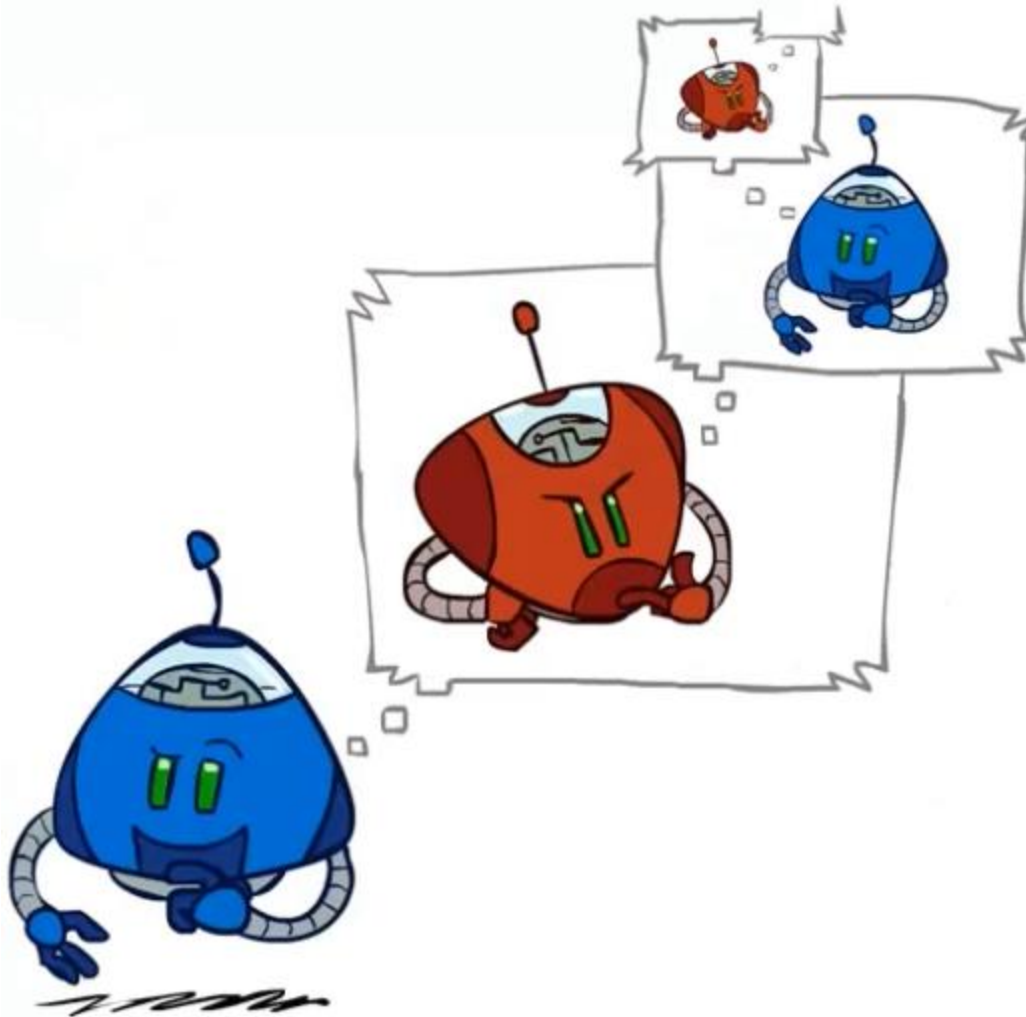
□ بازی‌های عمومی.

- عامل‌ها دارای سودمندی مجزا هستند.
- همکاری، بی‌تفاوتی و رقابت همگی ممکن هستند.
- بعداً در مورد این بازی‌ها بیشتر صحبت می‌کنیم.

□ بازی‌های مجموع صفر.

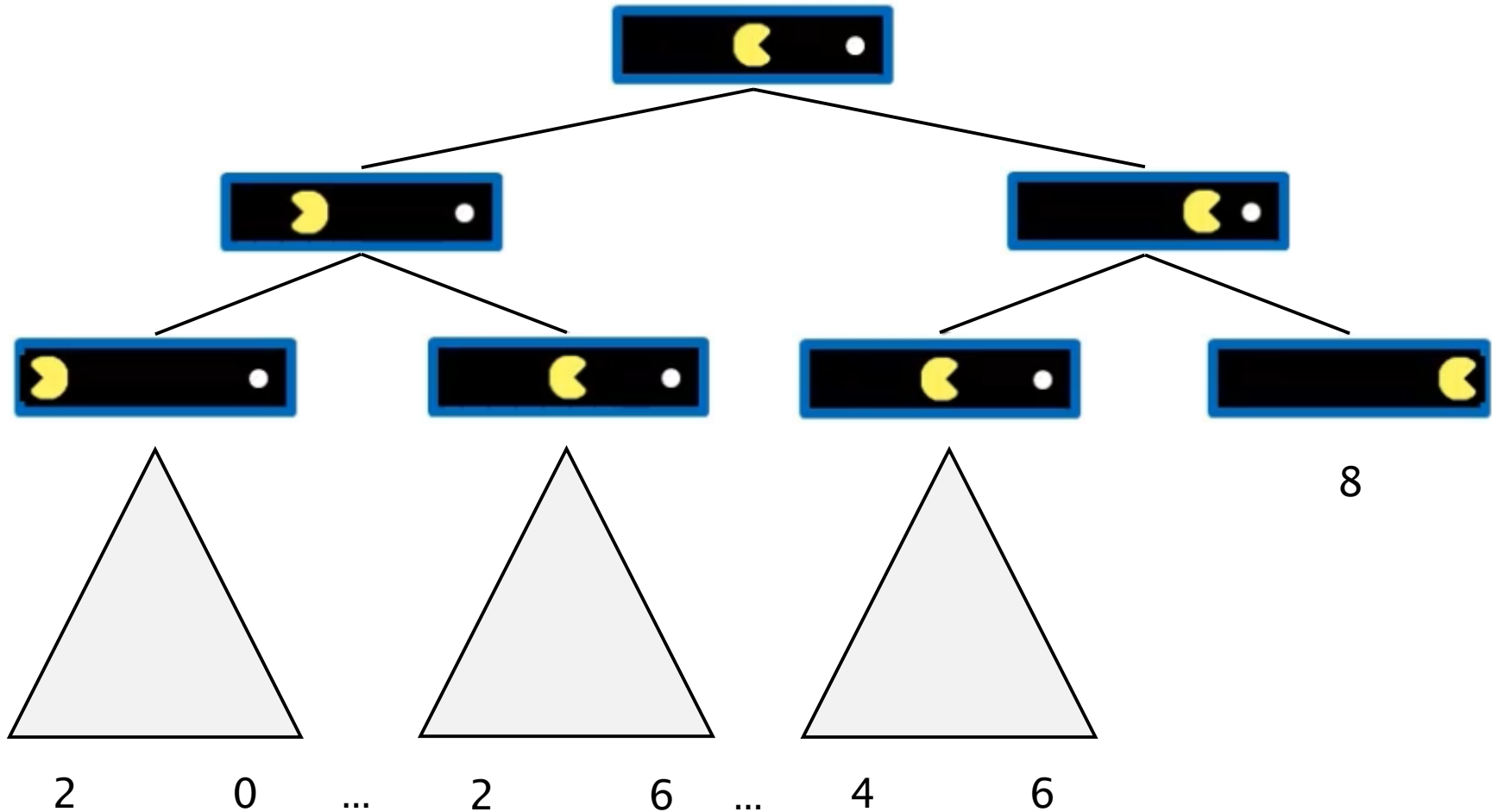
- عامل‌ها دارای سودمندی متضاد هستند.
- فرض کنید یک مقدار وجود دارد که یک بازیکن آن را بیشینه و دیگری آن را کمینه می‌سازد.
- خصمانه، اکیداً رقابتی

جستجوی رقابتی



درخت‌های تک عاملی

۹

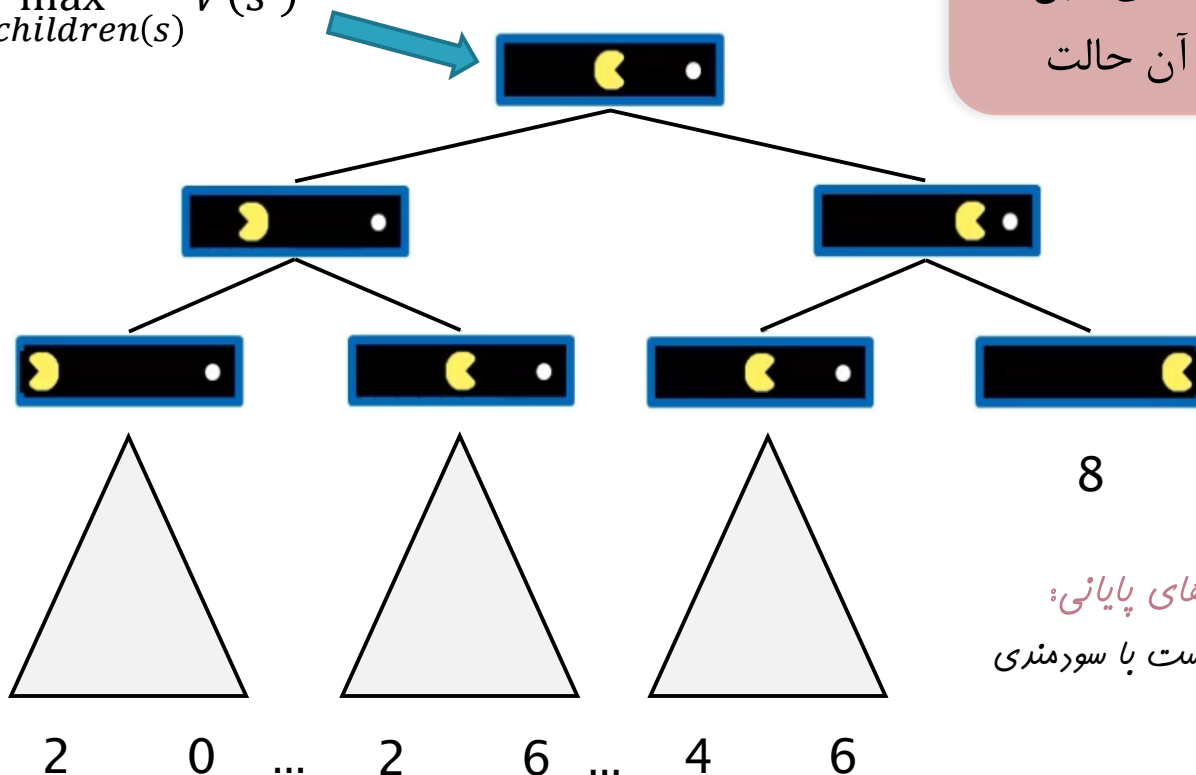


ارزش یک حالت

۱۰

حالت‌های غیر پایانی:

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



ارزش یک حالت:

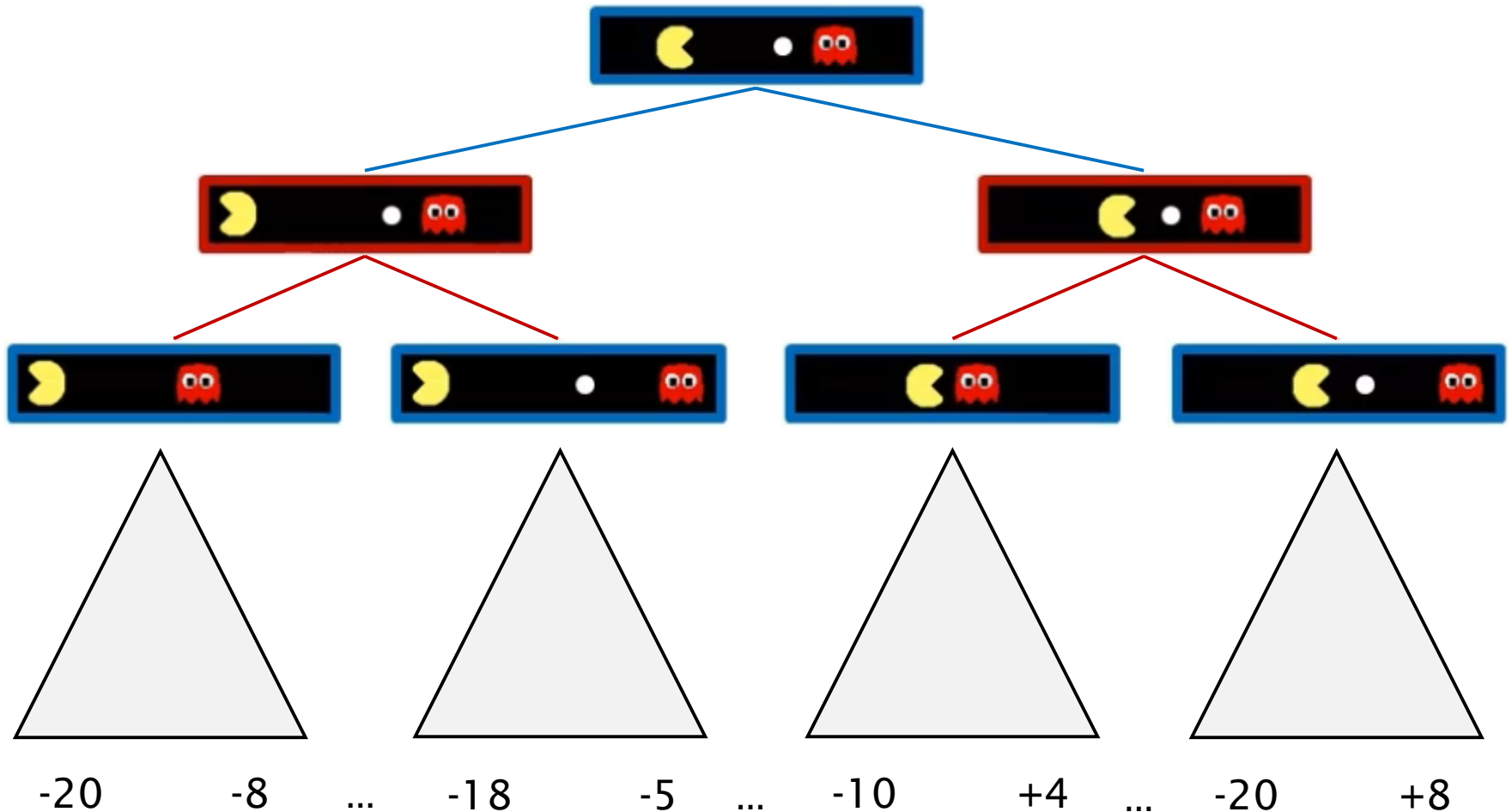
بهترین سودمندی قابل
دسترسی از آن حالت

حالت‌های پایانی:

ارزش برابر است با سودمندی

درخت بازی‌های رقابتی

۱۱



مقادیر Minimax

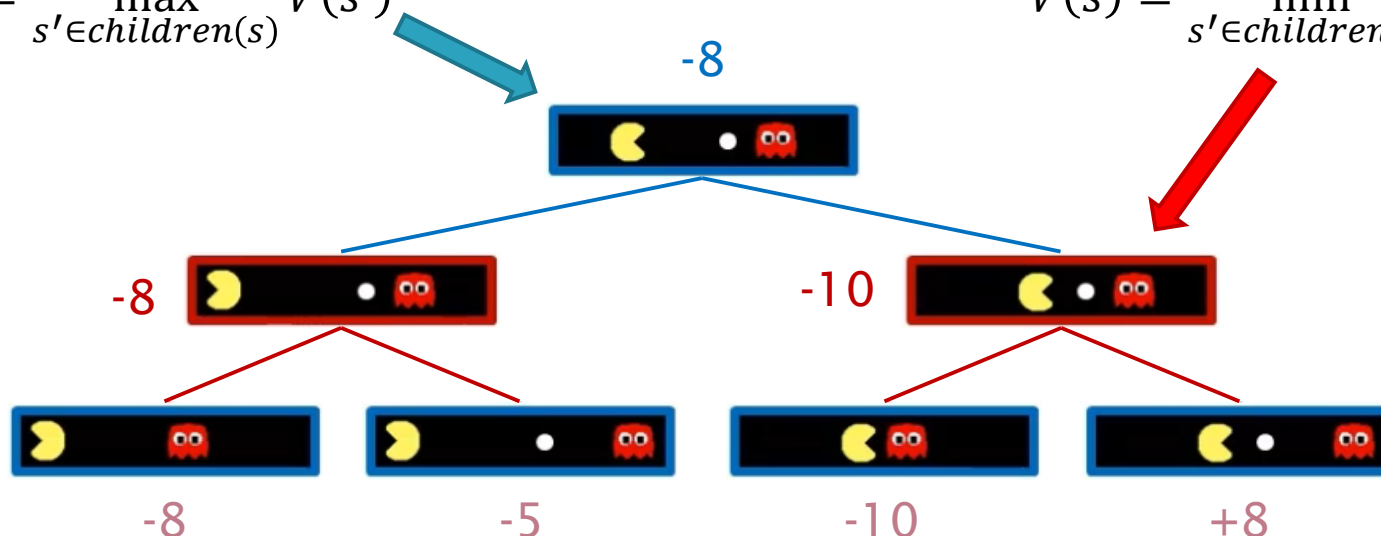
۱۲

حالت‌های تحت کنترل عامل:

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

حالت‌های تحت کنترل رقیب:

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



حالت‌های پایانی:

ارزش برابر است با سودمندی

درخت بازی Tic-Tac-Toe

۱۳



MAX (X)



MIN (O)



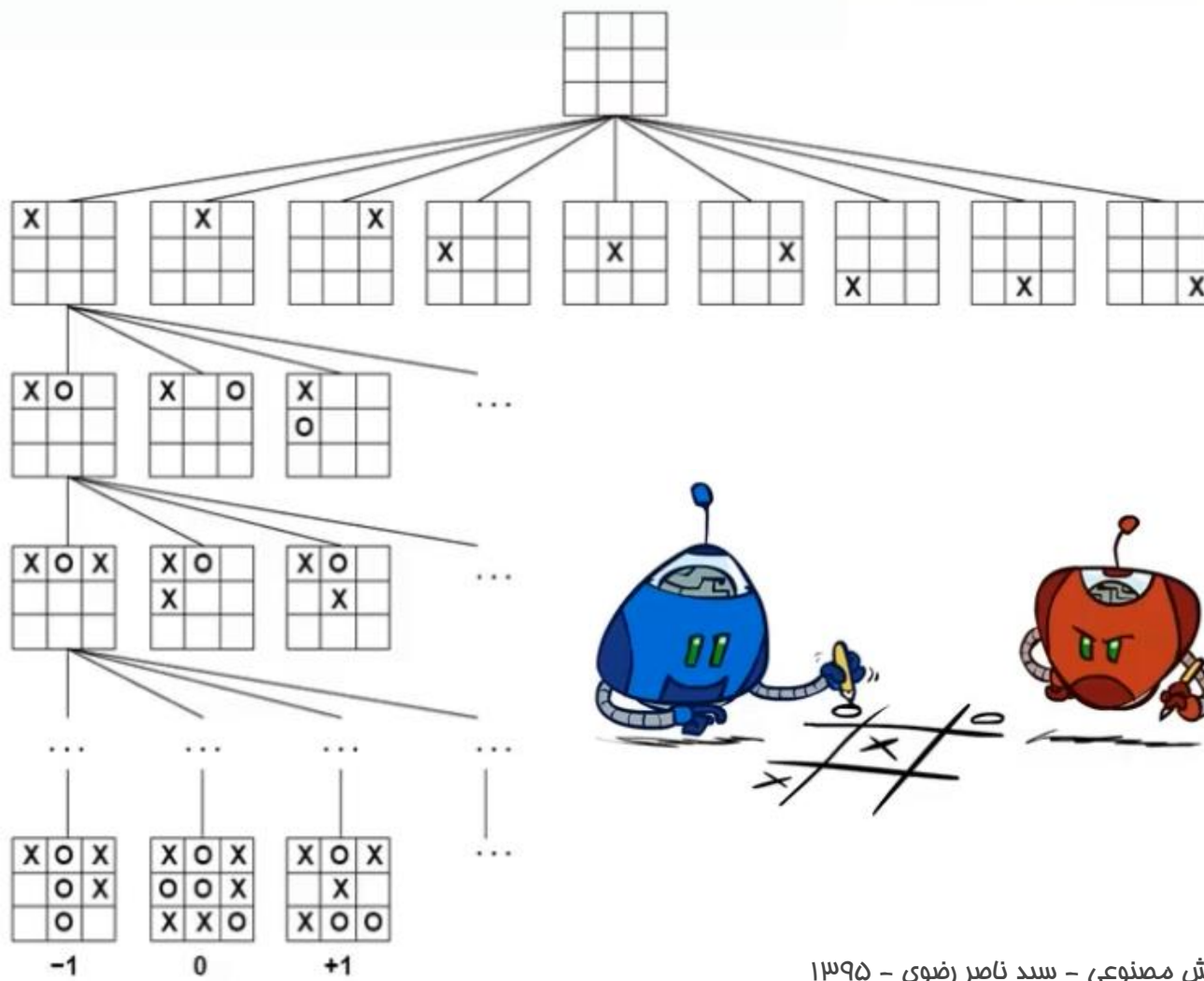
MAX (X)



MIN (O)

TERMINAL

Utility

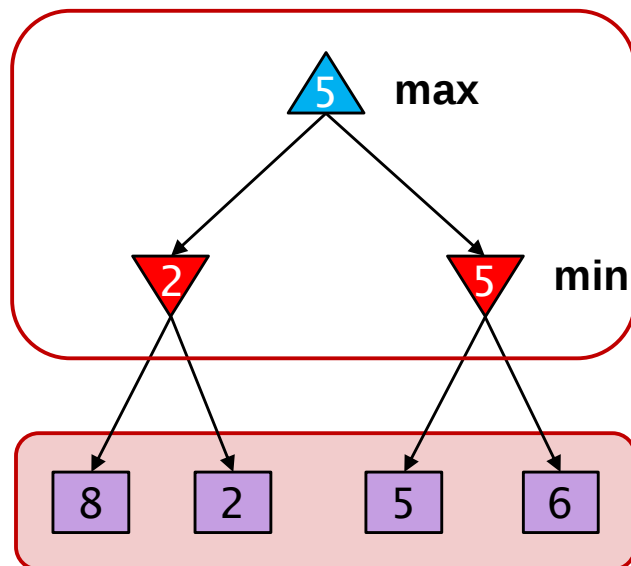


جستجوی رقابتی: الگوریتم Minimax

۱۴

مقادیر Minimax:

به صورت بازگشتی محاسبه می‌شوند



مقادیر حالت‌های پایانی:

به عنوان بخشی از بازی داده شده‌اند

□ بازی‌های قطعی، مجموع صفر.

□ دوز، شطرنج، چکرز

□ یک بازیکن سودمندی خود را بیشینه می‌سازد.

□ بازیکن رقیب سودمندی او را کمینه می‌سازد.

□ جستجوی Minimax:

□ یک درخت بازی

□ بازیکن‌ها به نوبت بازی می‌کنند.

□ محاسبه مقدار Minimax گره‌ها: بهترین

سودمندی قابل حصول در برابر یک رقیب
منطقی (بهینه).

پیاده‌سازی Minimax

۱۵

def max-value(state):

 initialize $v = -\infty$

 for each successor of state:

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

 return v

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

def min-value(state):

 initialize $v = +\infty$

 for each successor of state:

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

 return v

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

پیاده‌سازی Minimax

۱۶

def value(state):

if state is a TERMINAL: 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)

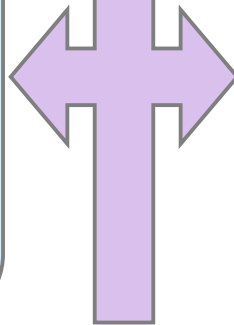
def max-value(state):

initialize $v = -\infty$

for each successor of state:

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

return v



def min-value(state):

initialize $v = +\infty$

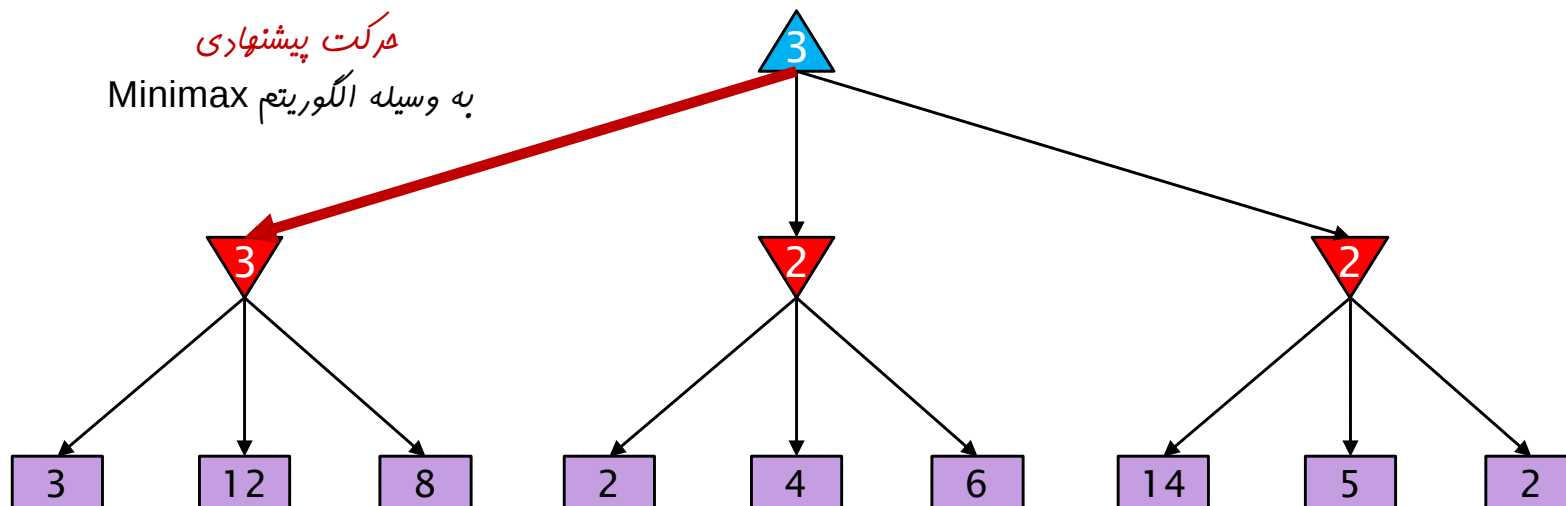
for each successor of state:

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

return v

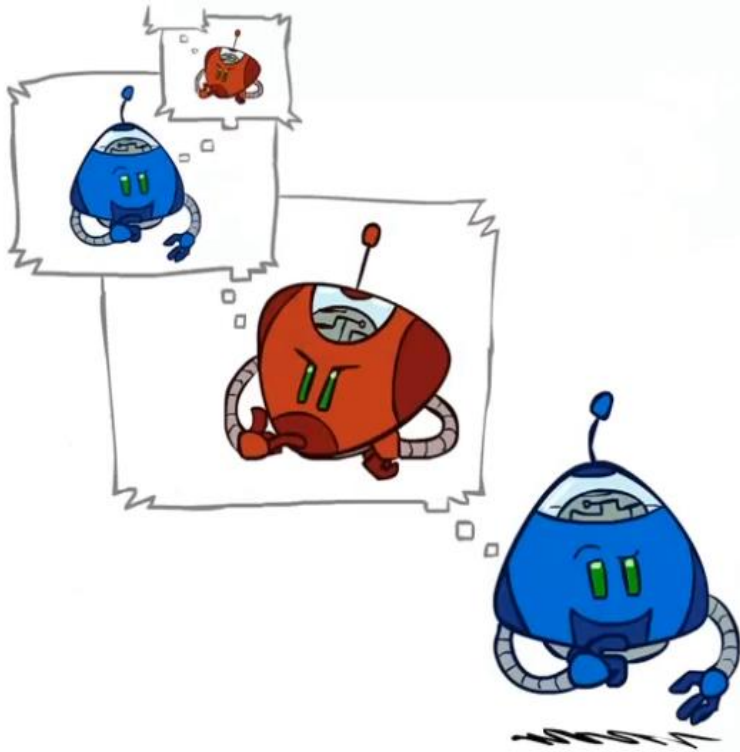
الگوریتم Minimax: مثال

۱۷



ارزیابی Minimax

۱۸



□ کامل. بله [به شرطی که درخت بازی محدود باشد]

□ بهینه. بله [به شرطی که رقیب هم بهینه بازی کند]

□ کارایی. درست همانند جستجوی عمقی

□ زمان: $O(b^m)$

□ حافظه: $O(bm)$

□ مثال. شطرنج $[b \approx 35, m \approx 100]$

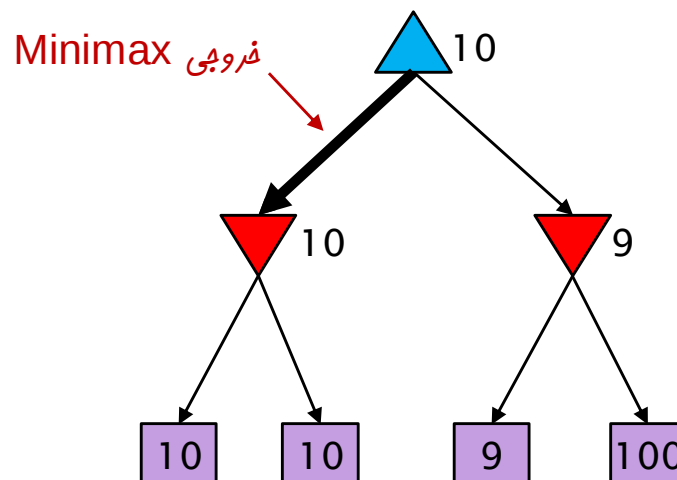
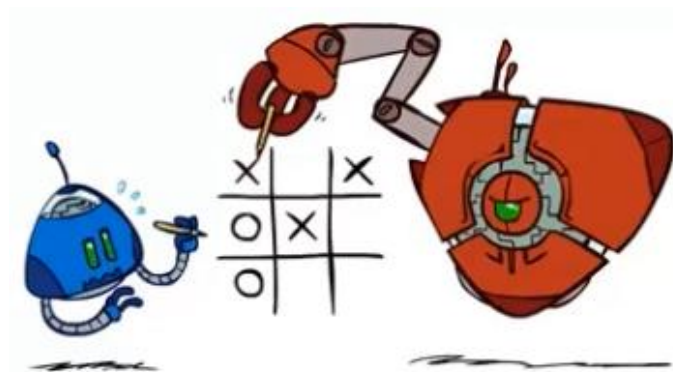
□ محاسبه راه حل دقیق عملی نیست.

□ س. آیا بررسی تمام درخت بازی لازم است؟

بهبندی Minimax

۱۹

□ بهینه. بله [به شرطی که رقیب هم بهینه بازی کند]

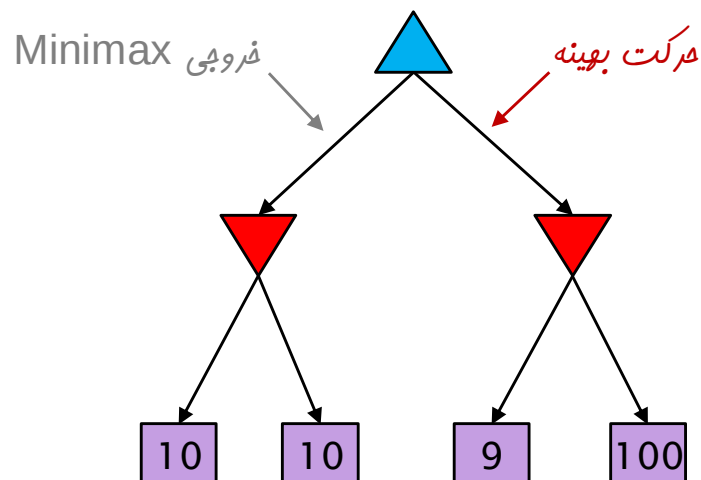
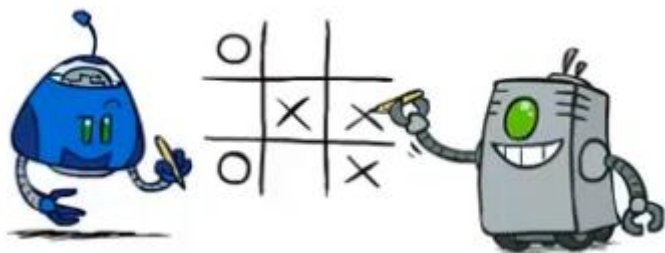


بهبندی Minimax

۲۰

□ بهینه. بله [به شرطی که رقیب هم بهینه بازی کند]

□ س. اگر رقیب بهینه بازی نکند، چه می‌شود؟

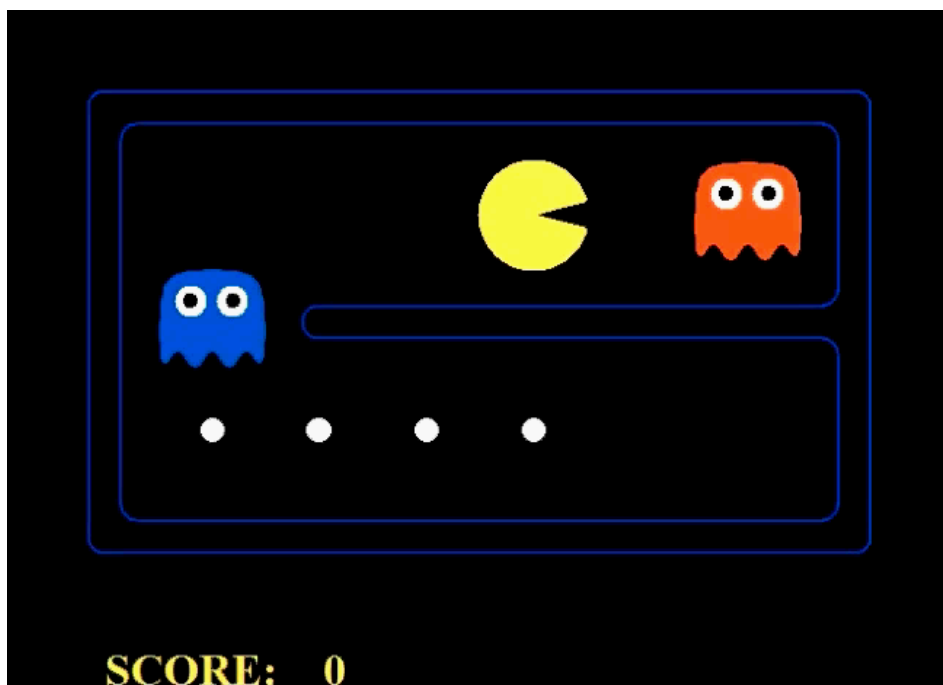


بازی در برابر رقبای بهینه

۲۱

□ س. در وضعیت نشان داده شده حرکت بهینه برای پکمن چیست؟ به شرطی که بدانیم رقبا بهینه بازی می کنند.

□ خوردن هر غذا: ۱۰ هر حرکت: ۱- خورده شدن پکمن: ۵۰۰-



```
python pacman.py -p MinimaxAgent -l trappedClassic -a depth=4 -z 2
```

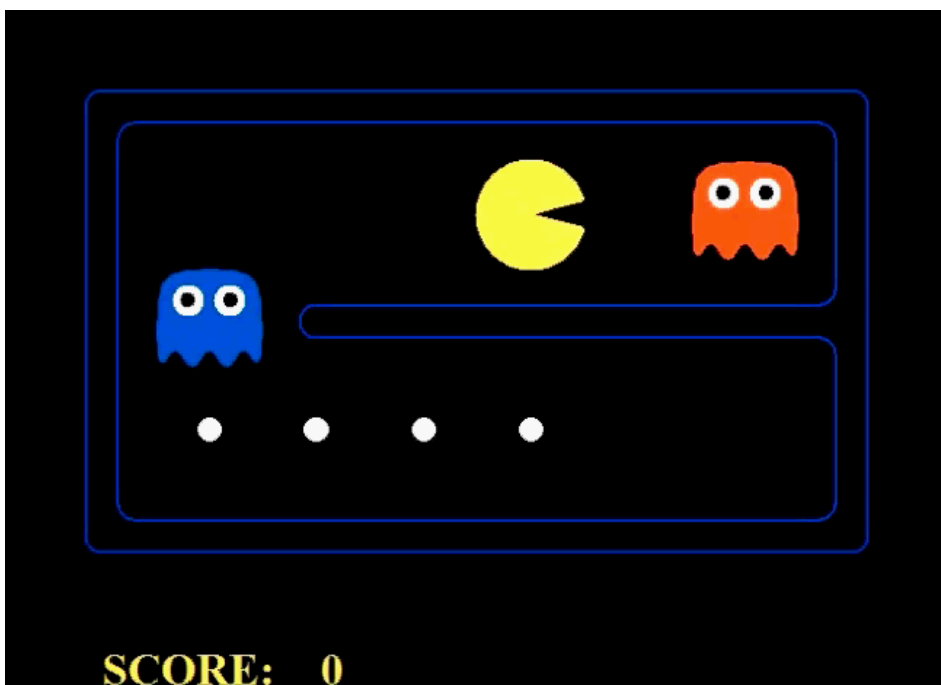
www.snrazavi.ir

بازی در برابر رقبای غیر بهینه

۲۲

□ س. در وضعیت نشان داده شده حرکت بهینه برای پکمن چیست؟ به شرطی که بدانیم رقبا به صورت تصادفی بازی می کنند.

□ خوردن هر غذا: ۱۰ هر حرکت: ۱- خورده شدن پکمن: ۵۰۰-



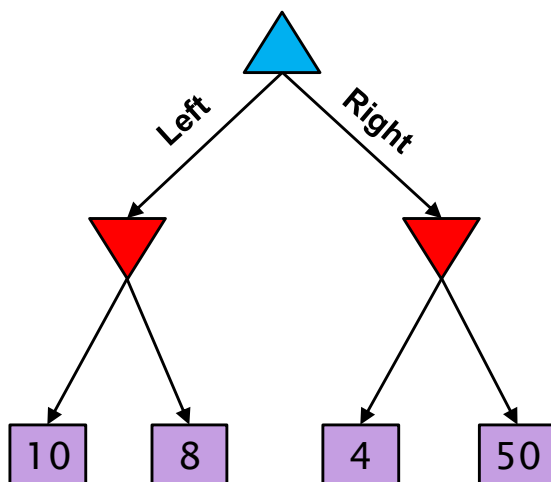
```
python pacman.py -p ExpectimaxAgent -l trappedClassic -a depth=4 -z 2
```

www.snrazavi.ir

پرسش کلاسی

۲۳

□ درخت بازی زیر را در نظر بگیرید:



□ مقدار minimax برای این درخت چیست؟

□ بر اساس استراتژی minimax بازیکن بیشینه‌ساز کدام عمل را انتخاب می‌کند؟

محدودیت منابع

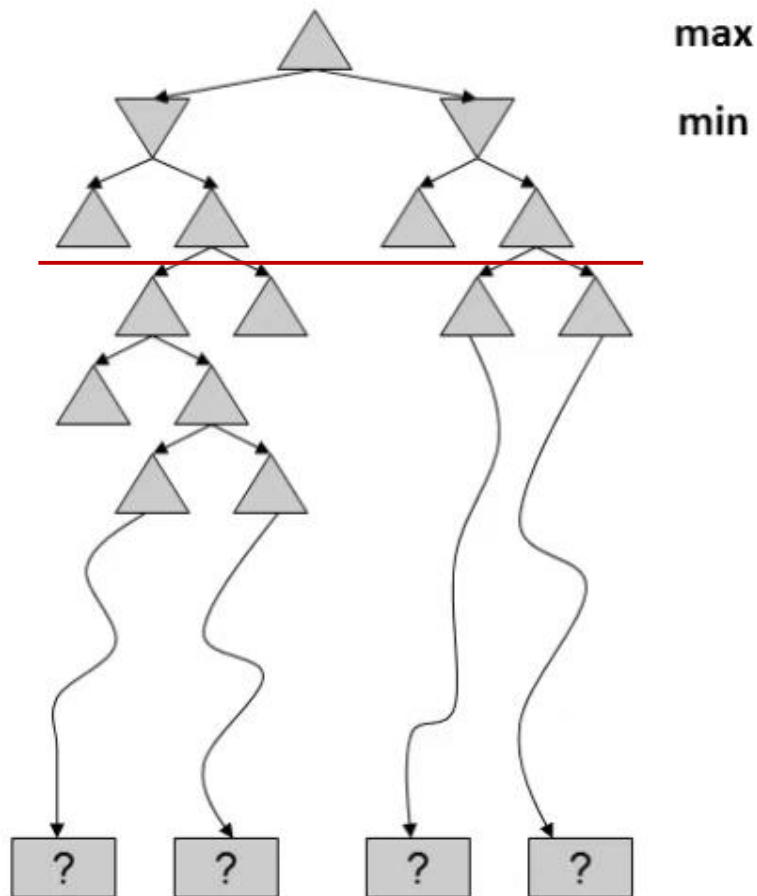
۲۴



محدودیت منابع

۲۵

❑ مشکل. در بازی‌های واقعی، نمی‌توان درخت را تا زمان رسیدن به برگ‌ها تولید نمود.



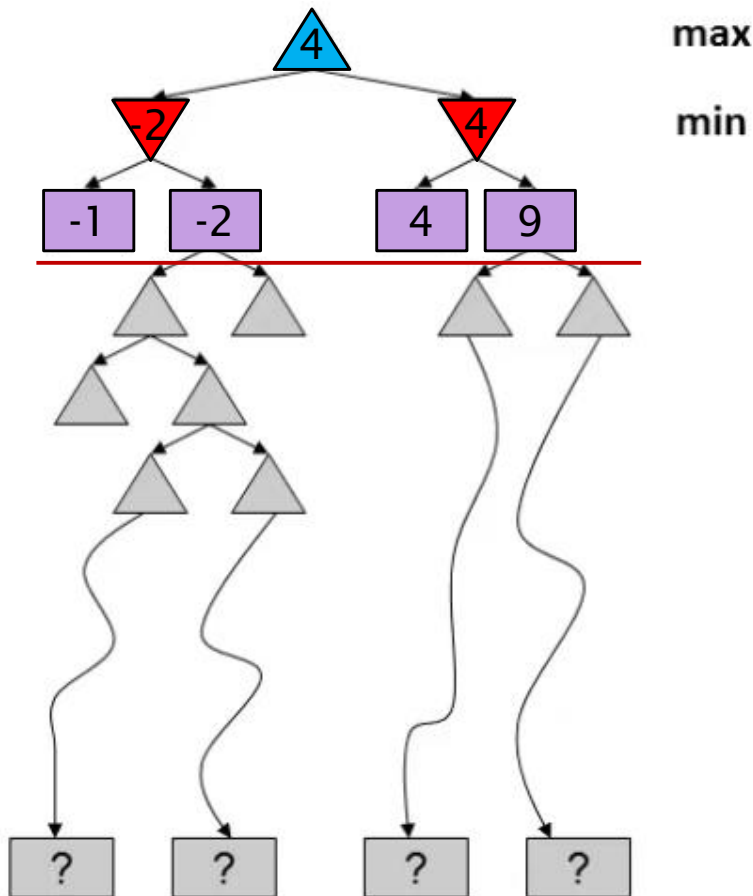
❑ راه‌حل. جستجوی عمقی با عمق محدود

❑ جستجو تا یک عمق محدود از درخت

محدودیت منابع

۲۶

❑ مشکل. در بازی‌های واقعی، نمی‌توان درخت را تا زمان رسیدن به برگ‌ها تولید نمود.



❑ راه‌حل. جستجوی عمقی با عمق محدود

❑ جستجو تا یک عمق محدود از درخت

❑ استفاده از یک تابع ارزیابی تخمینی برای

گره‌های غیرپایانی.

❑ مثال. شطرنج [$b \approx 35, m \approx 100$]

❑ برای هر حرکت ۱۰۰ ثانیه وقت داریم و

❑ می‌توانیم ۱۰۰۰۰ گره در ثانیه تولید کنیم.

❑ بنابراین می‌توانیم ۴ حرکت را پیش‌بینی کنیم.

محدودیت منابع

□ هیچ تضمینی برای بهینگی پاسخ وجود ندارد.

□ جستجوی تنها چند لایه دیگر می تواند تفاوت زیادی در نتیجه ایجاد کند.

□ اگر بخواهیم به ازای هر مدت زمان دلخواه حتماً یک راه حل پیدا کنیم، می توانیم از جستجوی عمیق کننده تکراری استفاده کنیم. [یک الگوریتم هر زمانی]

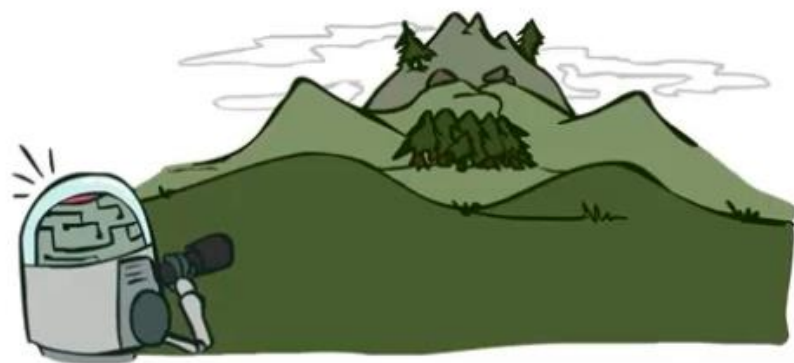
اهمیت عمق جستجو

۲۸



□ توابع ارزیابی هرگز عالی و بدون نقص نیستند.

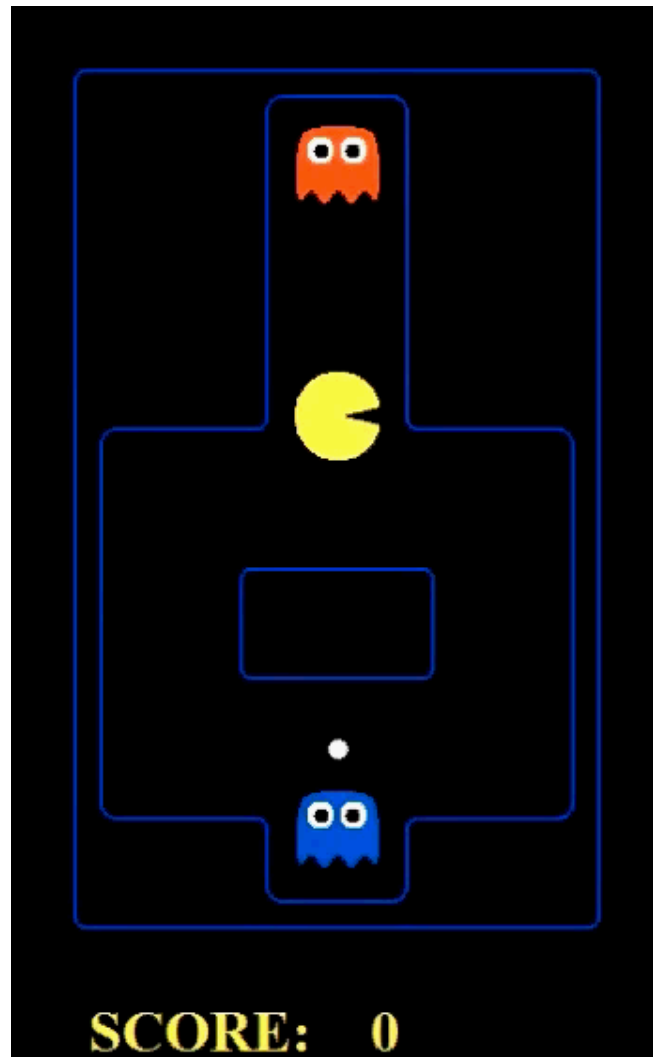
□ هر چقدر بیشتر در عمق پیش برویم، کیفیت تابع ارزیابی اهمیت کمتری پیدا می‌کند.



□ یک مثال مهم از موازنه میان پیچیدگی ویژگی‌ها و پیچیدگی محاسبات!

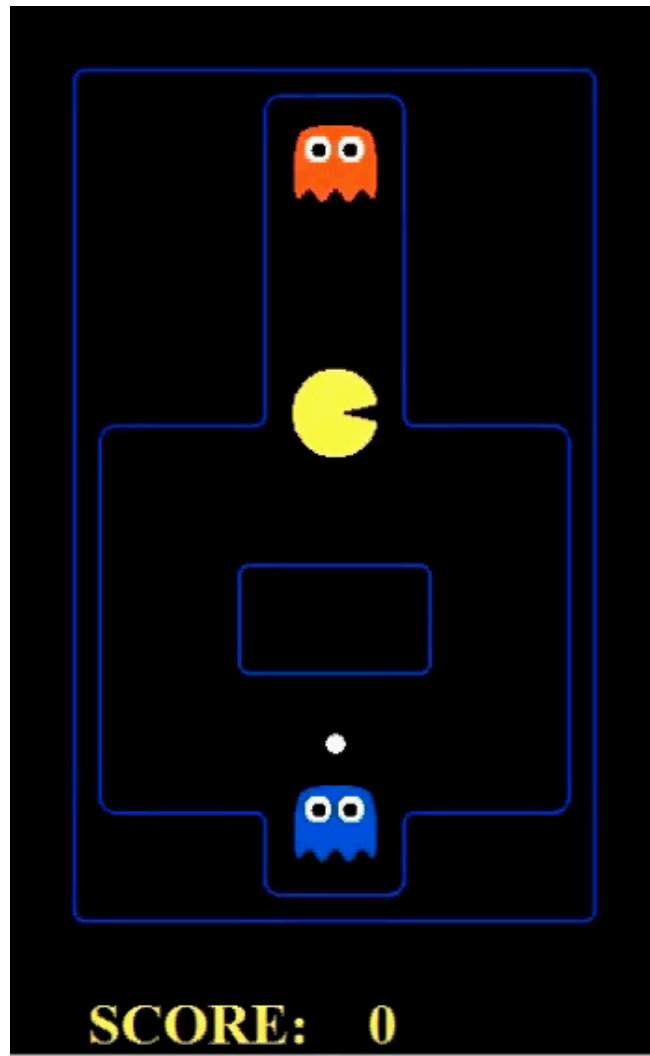
مثال: جستجوی عمقی با محدوده عمقی ۲

۲۹



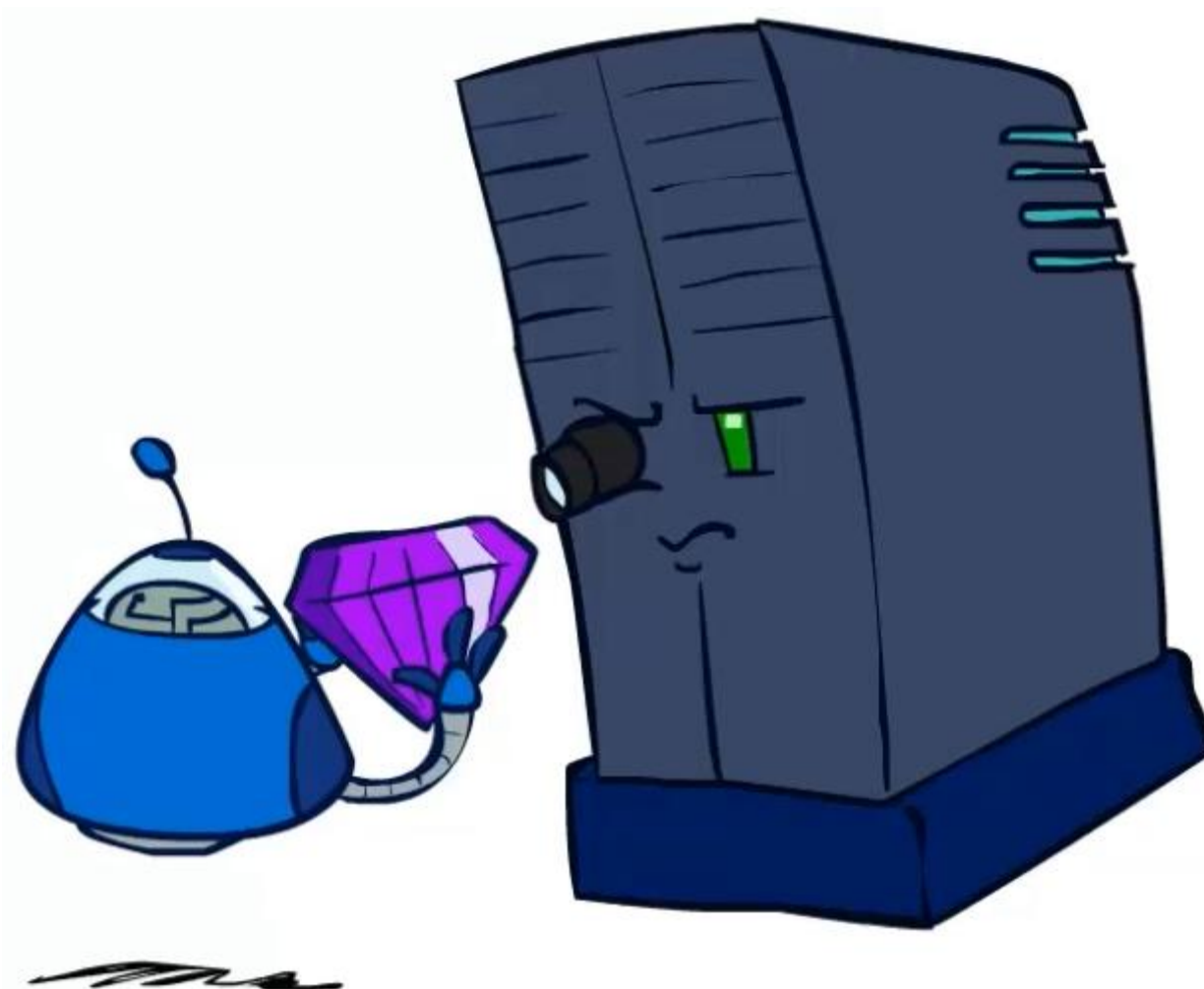
مثال: جستجو بدون محدود کردن عمق

۳۰



توابع ارزیابی

۳۱



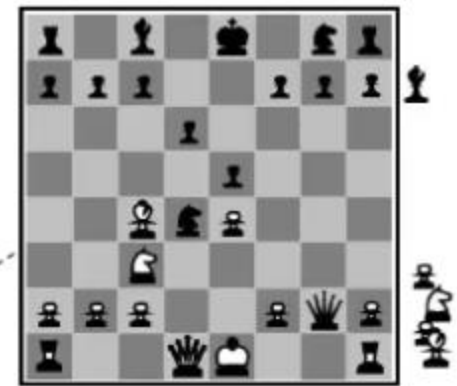
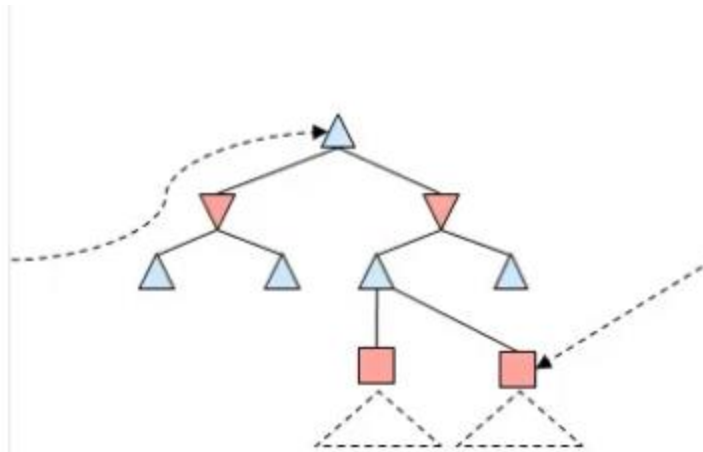
توابع ارزیابی

۳۲

□ توابع ارزیابی. ارزیابی گره‌های غیر پایانی در جستجوی عمقی محدود.



نوبت سیاه. سفید نسبتاً بهتر است



نوبت سفید. سیاه برنده است

□ تابع ایده‌آل. مقدار واقعی minimax موقعیت داده شده را برمی‌گرداند.

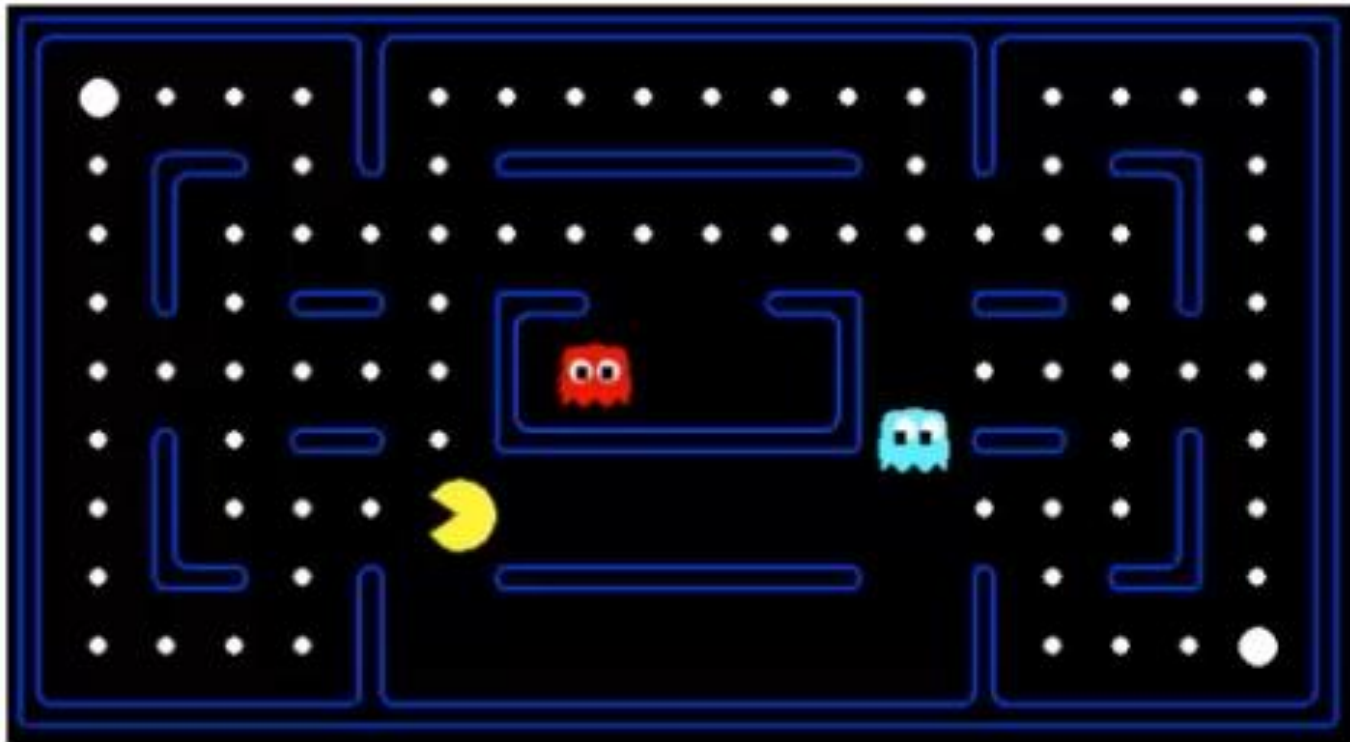
□ در عمل. معمولاً از یک مجموع خطی وزن‌دار از ویژگی‌ها استفاده می‌شود:

$$Eval(s) = w_1f_1(s) + w_2f_2(s) + \dots + w_nf_n(s)$$

تعداد وزیرهای سفید - تعداد وزیرهای سیاه

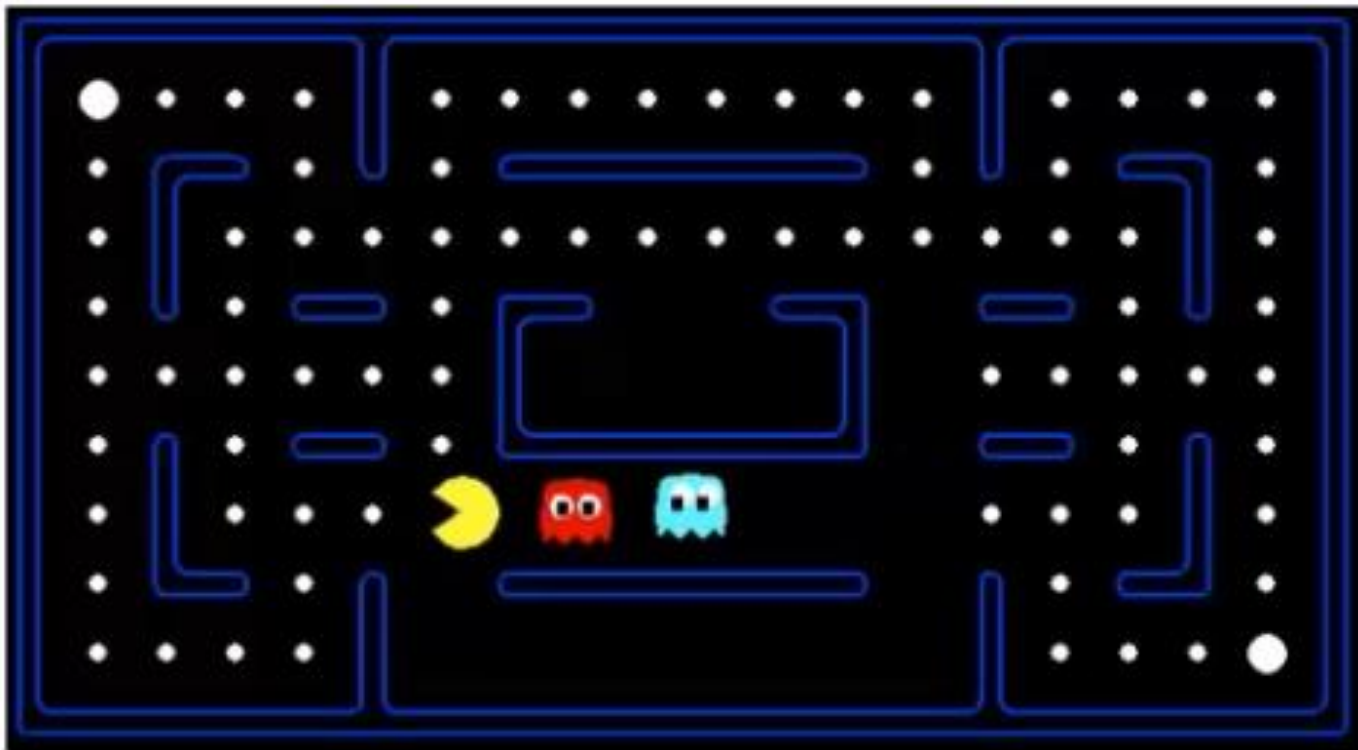
ارزیابی برای بازی پکمن

۳۳



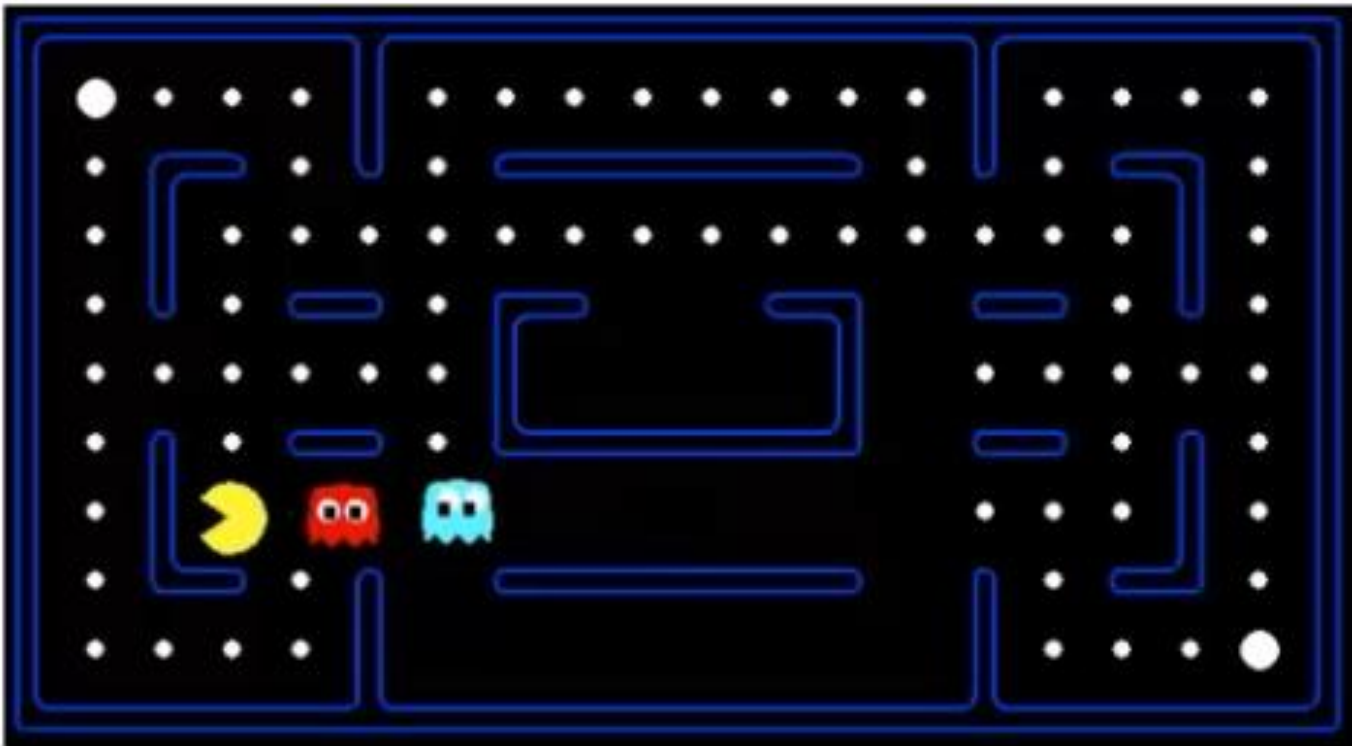
ارزیابی برای بازی پکمن

۳۴



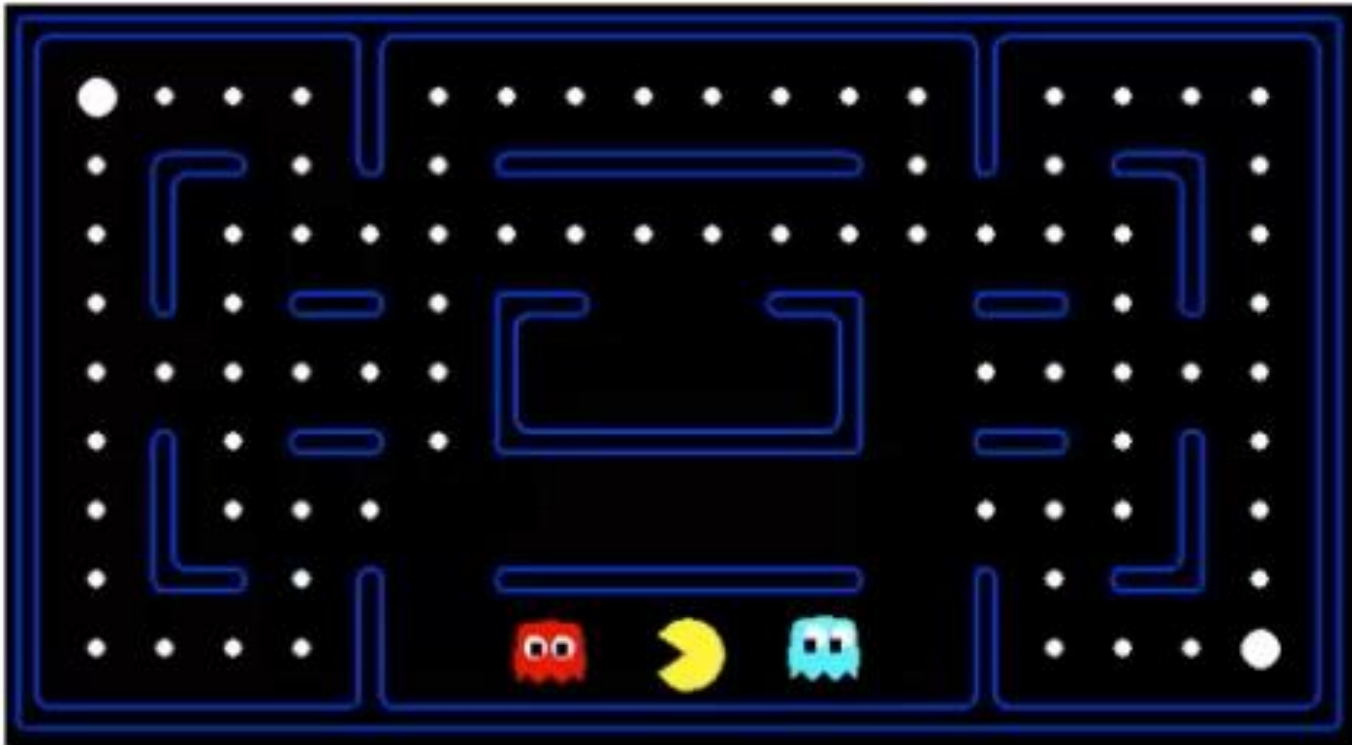
ارزیابی برای بازی پکمن

۳۵



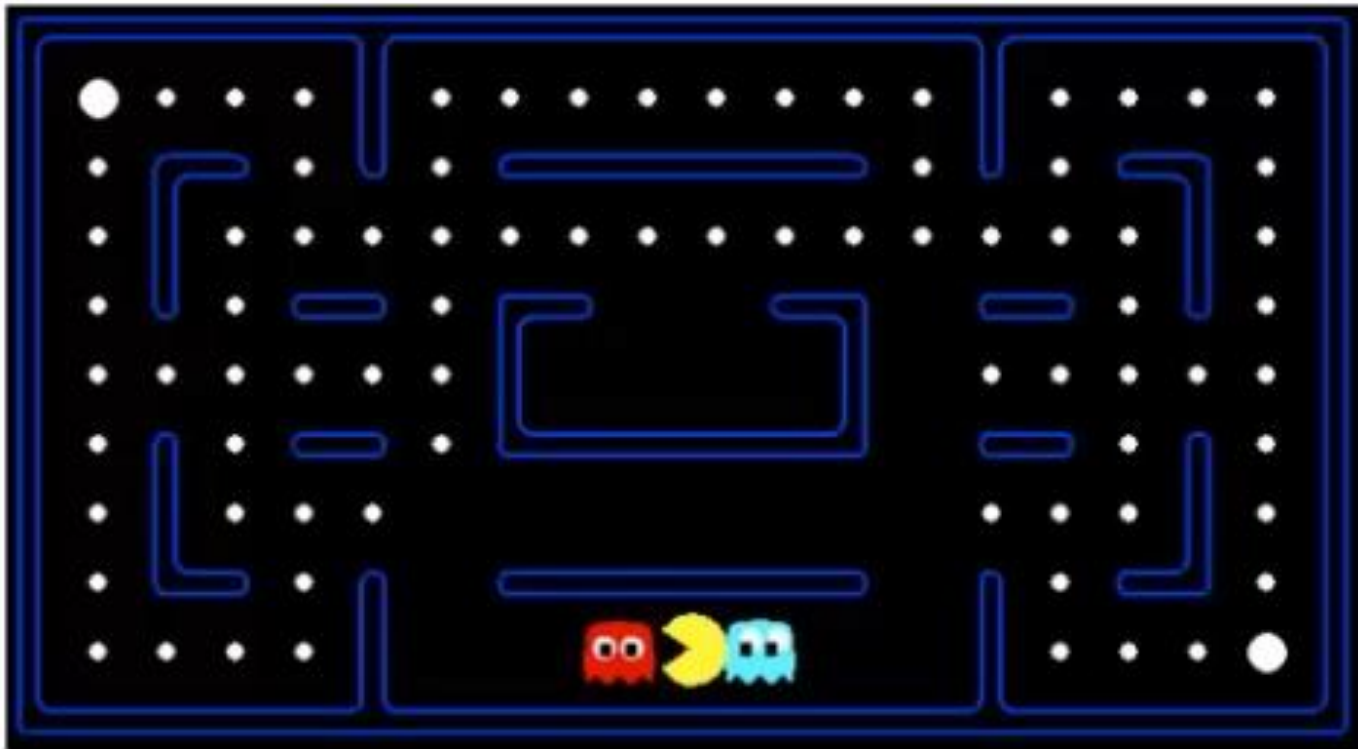
ارزیابی برای بازی پکمن

۳۶



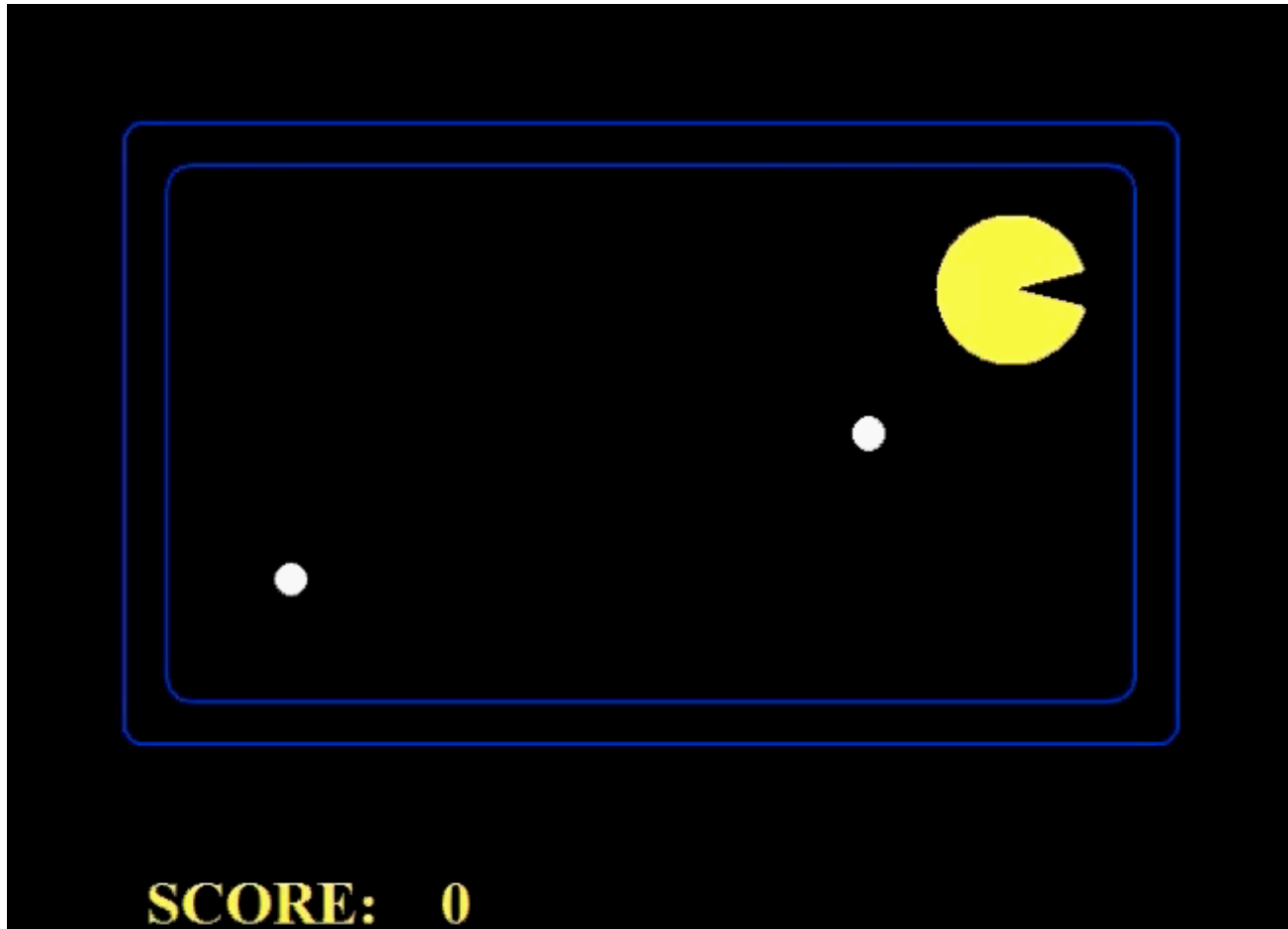
ارزیابی برای بازی پکمن

۳۷



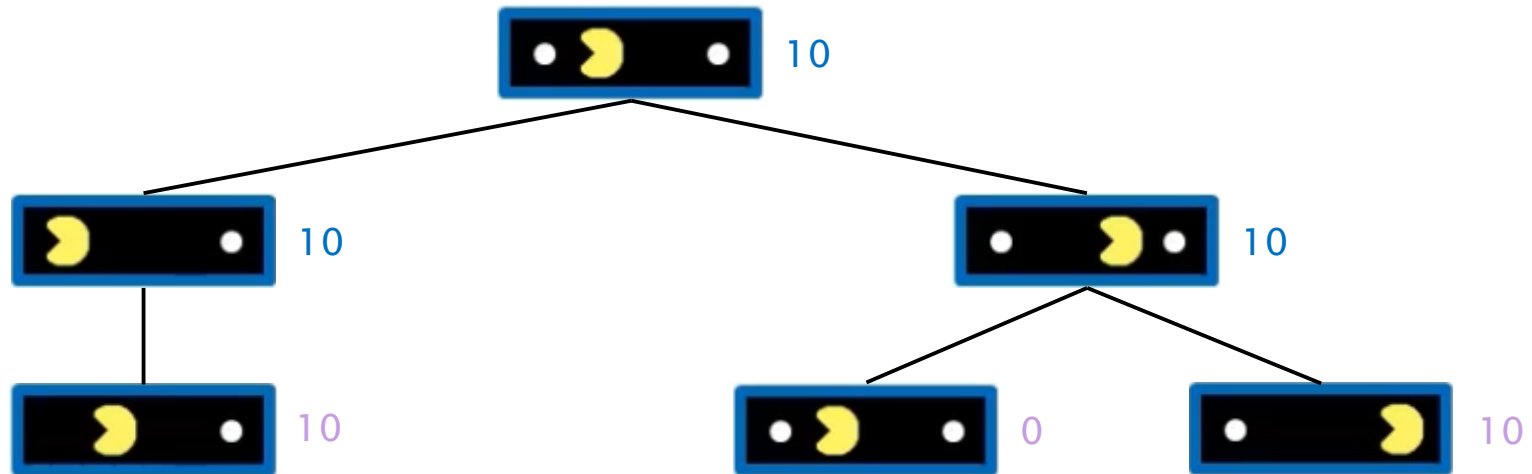
ارزیابی برای بازی پکمن: اثر کوبیدگی

۳۸



بررسی علت گرسنه ماندن پکمن

۳۹

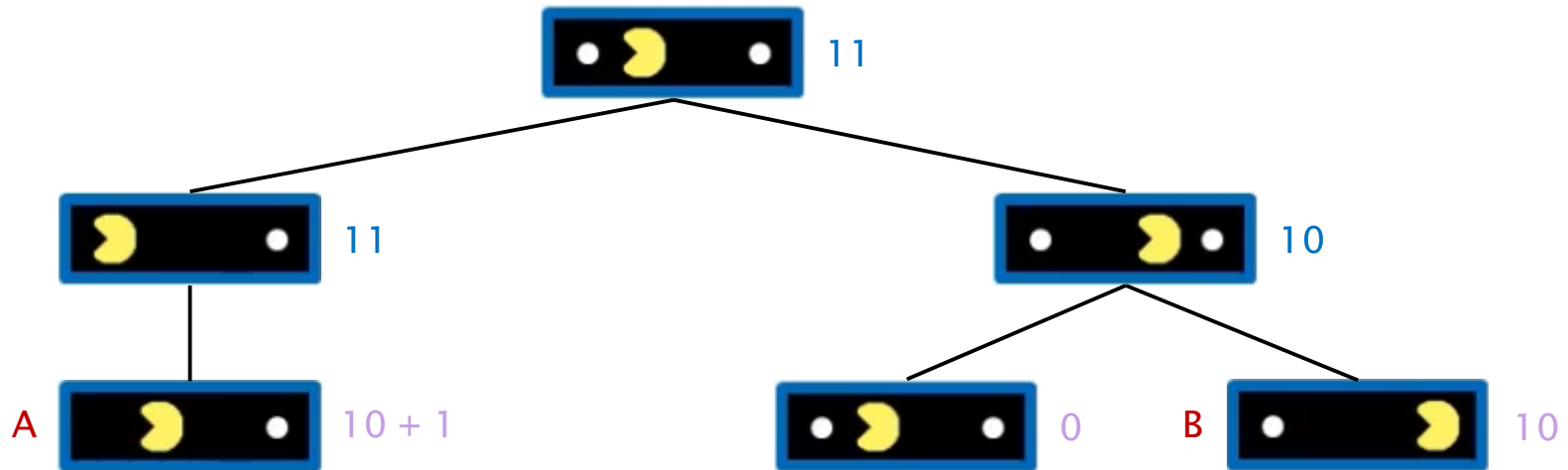


□ خطر عامل‌هایی که مجدداً برنامه‌ریزی می‌کنند!

- عامل می‌داند اگر اکنون غذا بخورد امتیازش افزایش می‌یابد: $[W, E]$
- همچنین می‌داند که اگر غذا را بعداً بخورد، به همان اندازه امتیازش افزایش می‌یابد: $[E, W]$
- بعد از خوردن غذا دیگر فرصتی برای افزایش امتیاز وجود ندارد (در افق دید برابر با ۲)
- بنابراین منتظر ماندن و خوردن غذا به یک اندازه امتیاز دارند: بنابراین ممکن است عامل به راست برود و سپس در دور بعدی برنامه‌ریزی به چپ برود!

تغییر تابع ارزیابی

۴۰

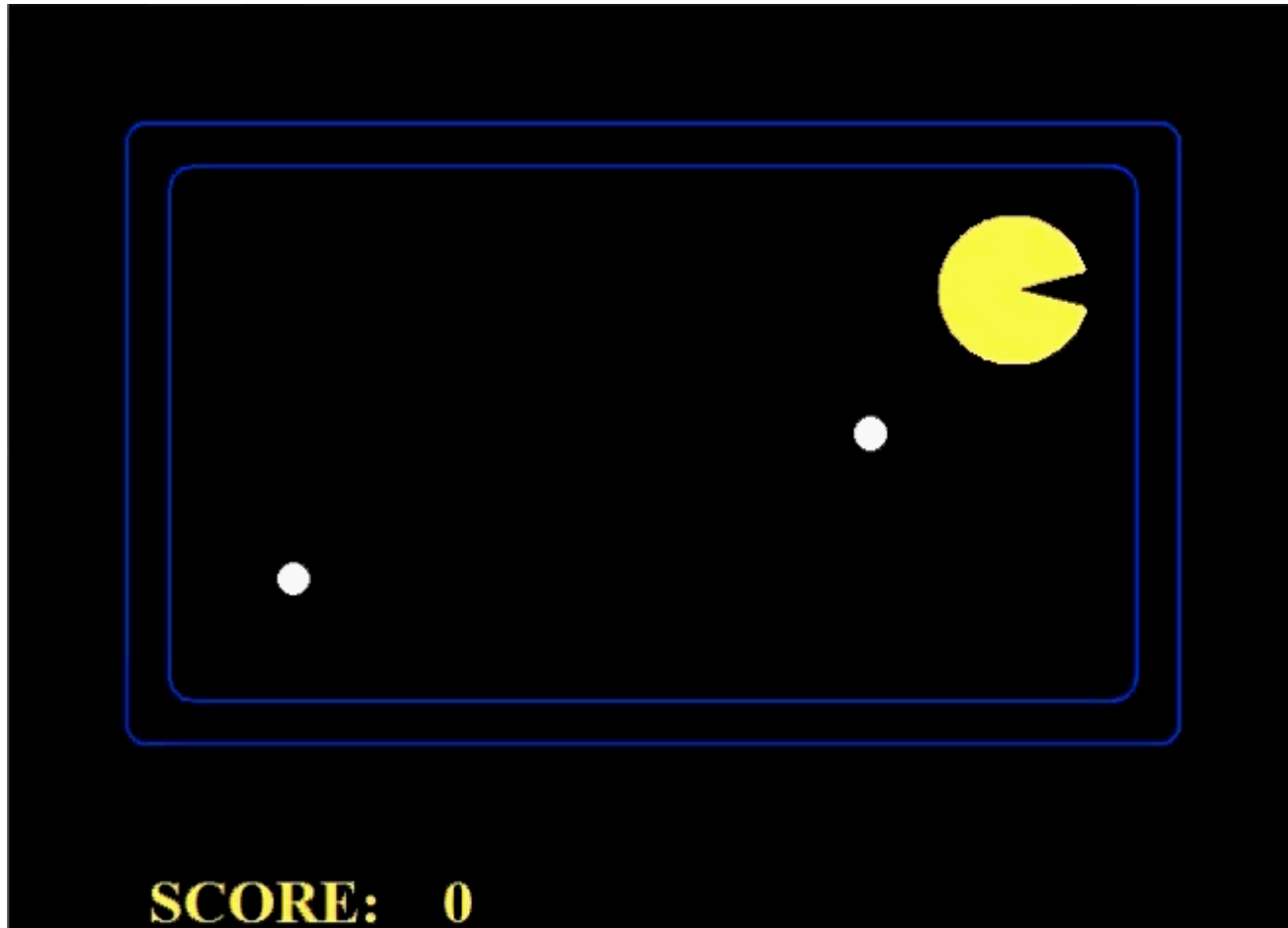


□ رفع مشکل.

- از نظر تابع ارزیابی قبلی، حالت‌های A و B به یک اندازه خوب بودند.
- اما حالت A کمی بهتر از حالت B است. [زیرا در این حالت عامل به غذا نزدیک‌تر است]
- اگر تابع ارزیابی به گونه‌ای اصلاح شود که برای حالت A امتیاز بیشتری نسبت به حالت B در نظر بگیرد، عامل به سمت چپ می‌رود و به این ترتیب مشکل گرسنه ماندن عامل حل می‌شود.

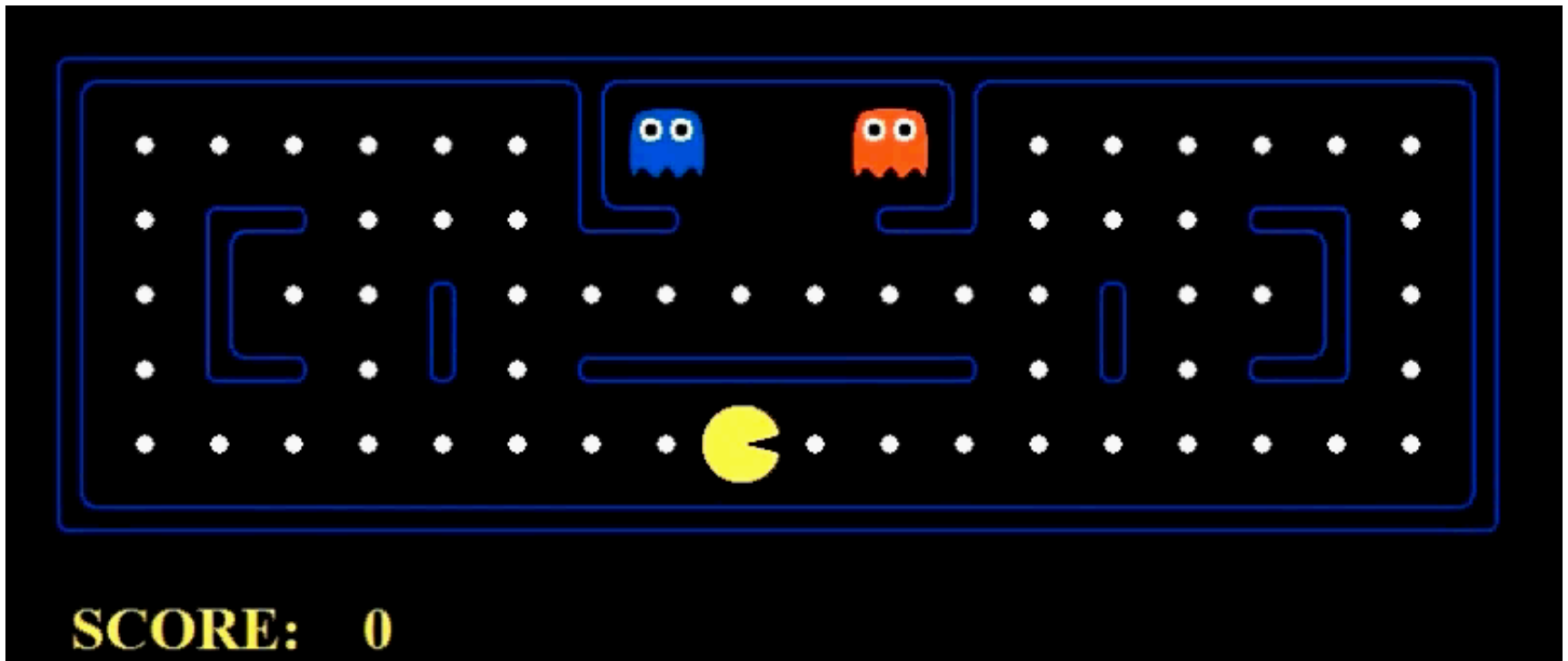
ارزیابی برای بازی پکمن: حل مشکل کوبیدگی

۴۱



ارزیابی با وجود رقیب

۴۲

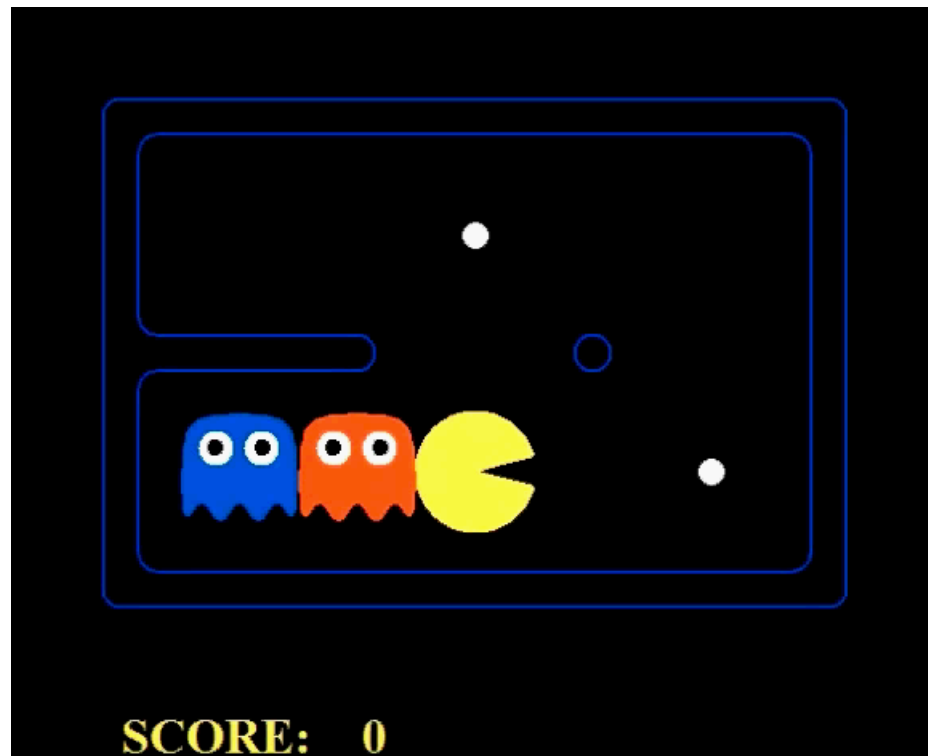


همکاری میان عامل‌ها بدون برنامه‌نویسی

۴۳

□ همکاری بدون برنامه‌نویسی.

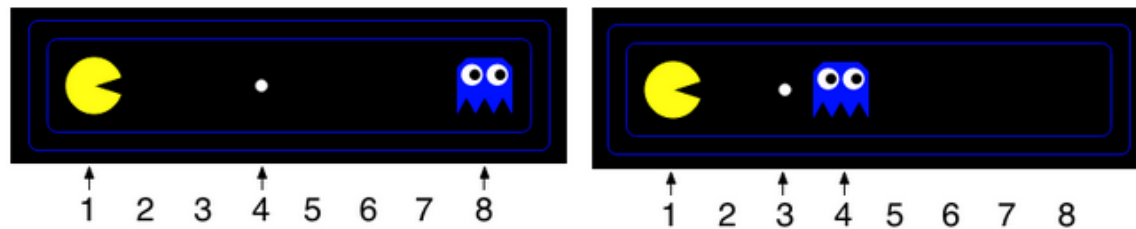
- هر یک از عامل‌های رقیب به طور جداگانه برنامه‌ریزی می‌کنند.
- اما چون دارای هدف مشترکی هستند، بین آنها همکاری به وجود می‌آید.



پرسش کلاسی

۴۴

□ کدام یک از توابع ارزیابی زیر به حالت سمت چپ امتیاز بالاتری نسبت می‌دهد؟



□ عکس فاصله پکمن تا نزدیک‌ترین غذا

□ فاصله پکمن تا نزدیک‌ترین روح

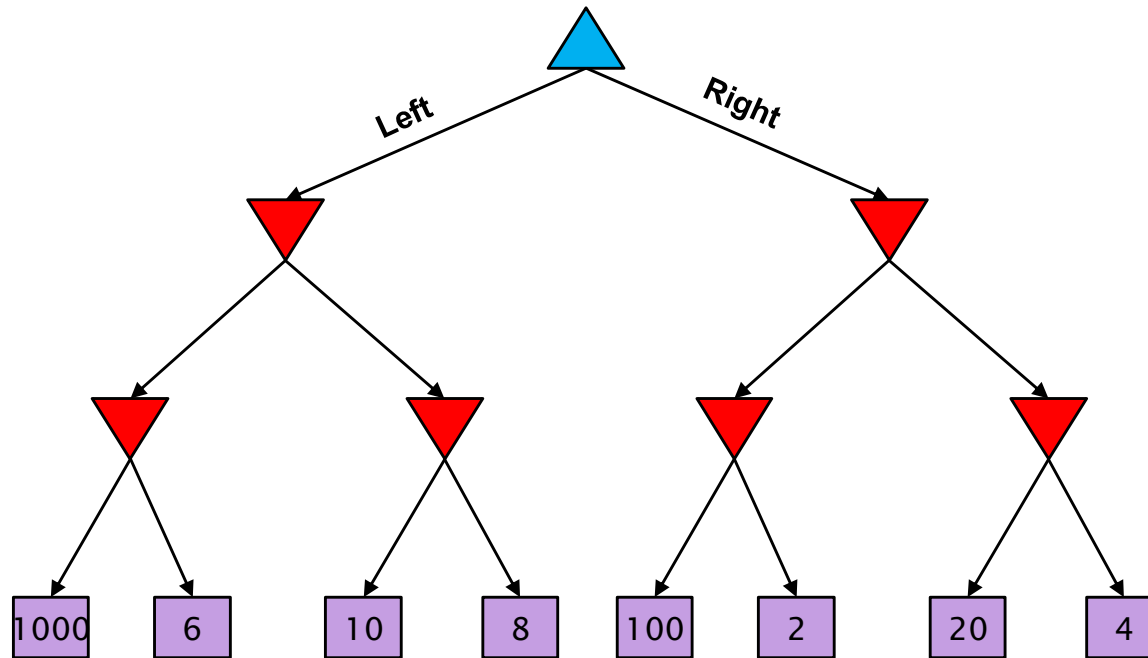
□ فاصله پکمن تا نزدیک‌ترین روح + عکس فاصله پکمن تا نزدیک‌ترین غذا

□ فاصله پکمن تا نزدیک‌ترین روح + (عکس فاصله پکمن تا نزدیک‌ترین غذا) $\times 1000$

پرسش کلاسی

۴۵

□ درخت بازی زیر را در نظر بگیرید:

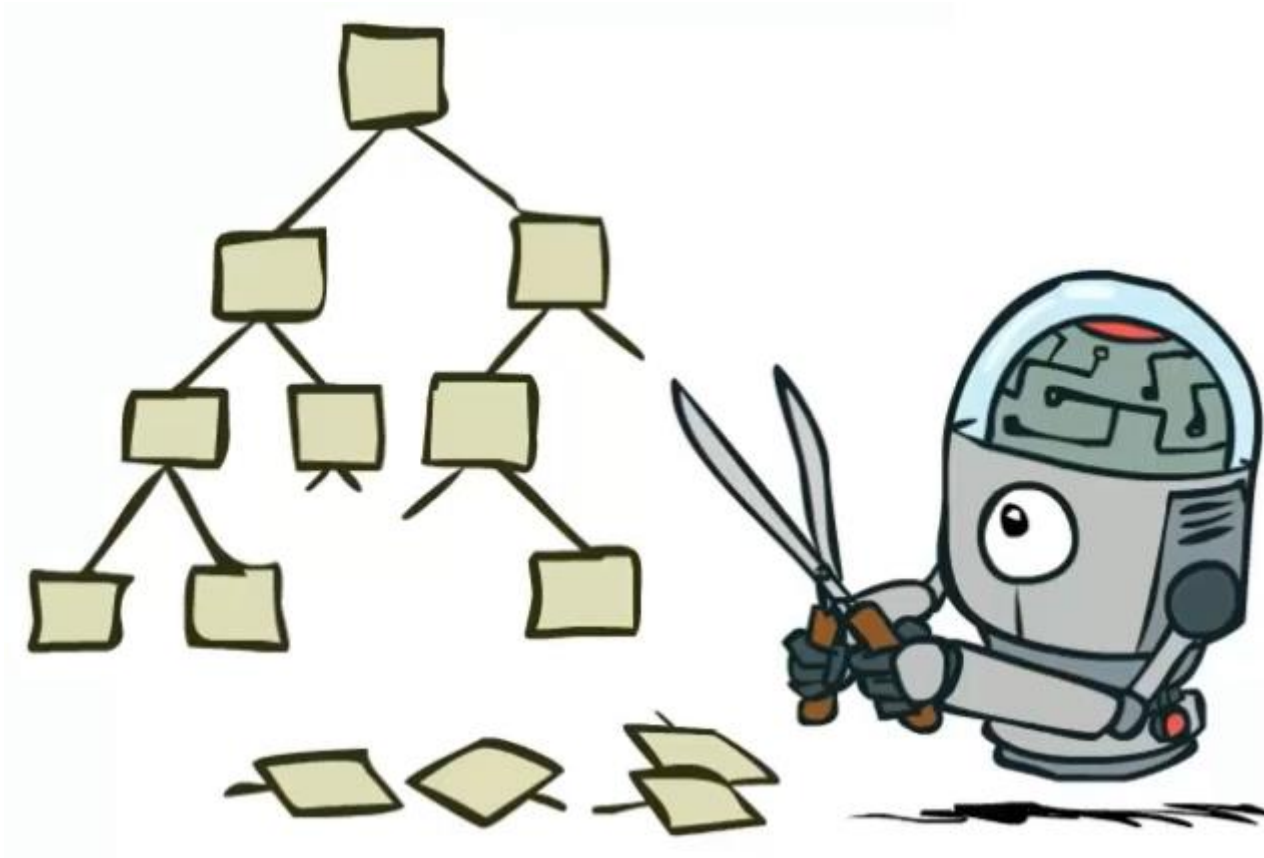


□ مقدار minimax برای این درخت چیست؟

□ بر اساس استراتژی minimax بازیکن بیشینه‌ساز کدام عمل را انتخاب می‌کند؟

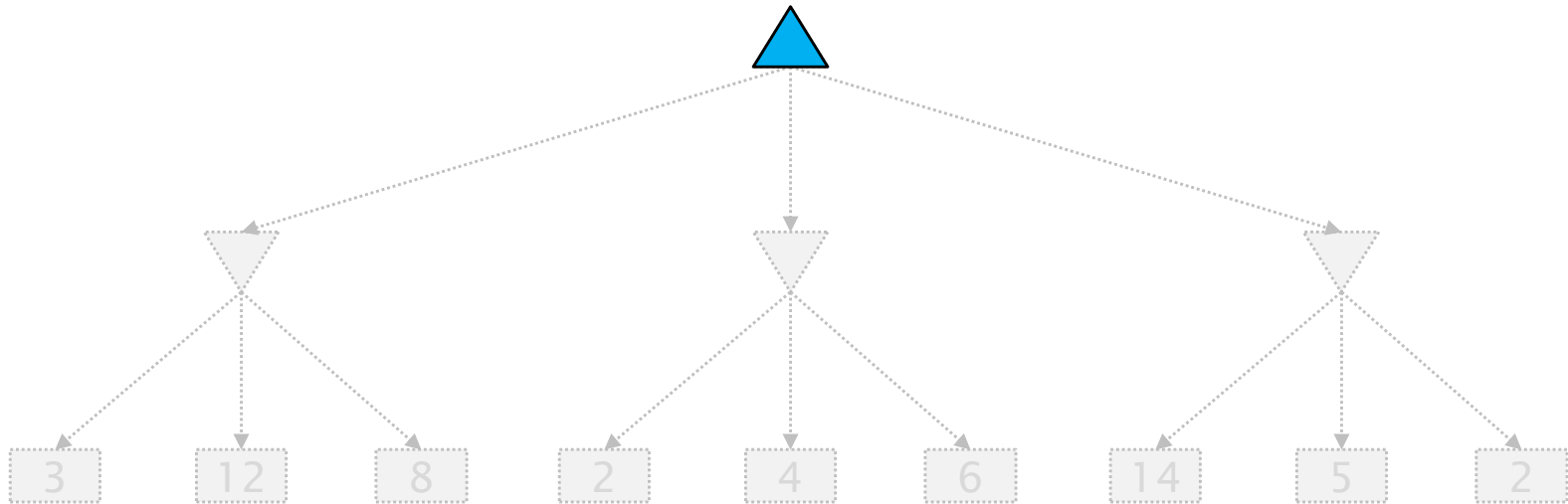
هرس کردن درخت بازی

۴۶



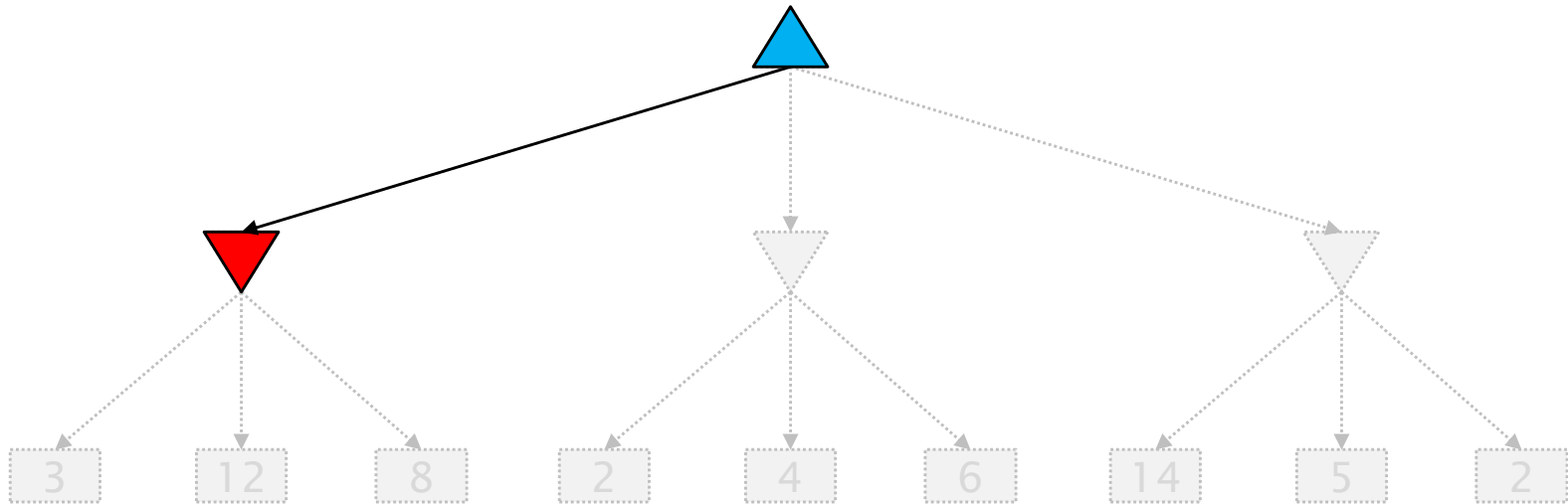
الگوریتم Minimax: یک نگاه دقیق‌تر

۴۷



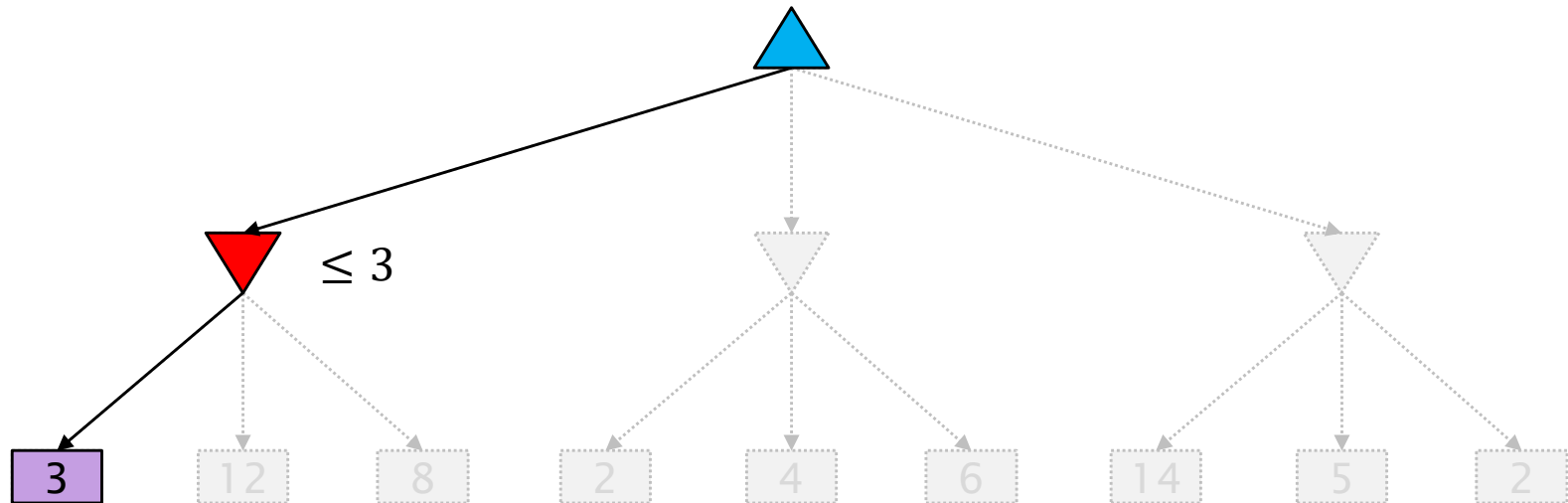
الگوریتم Minimax: یک نگاه دقیق‌تر

۴۸



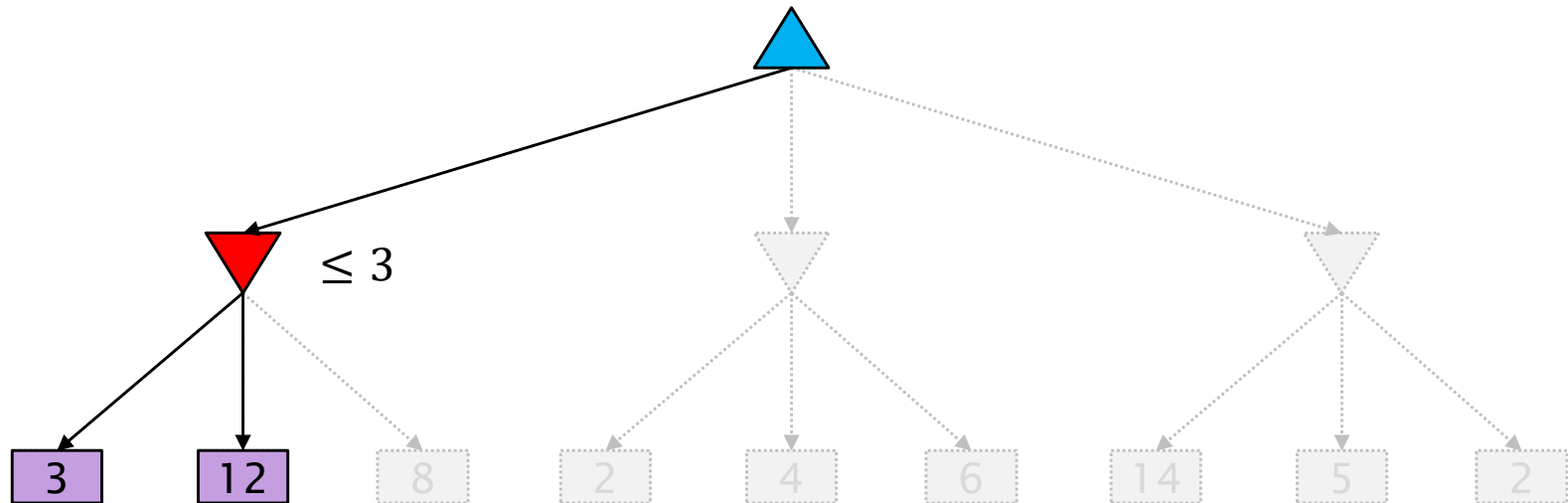
الگوریتم Minimax: یک نگاه دقیق‌تر

۴۹



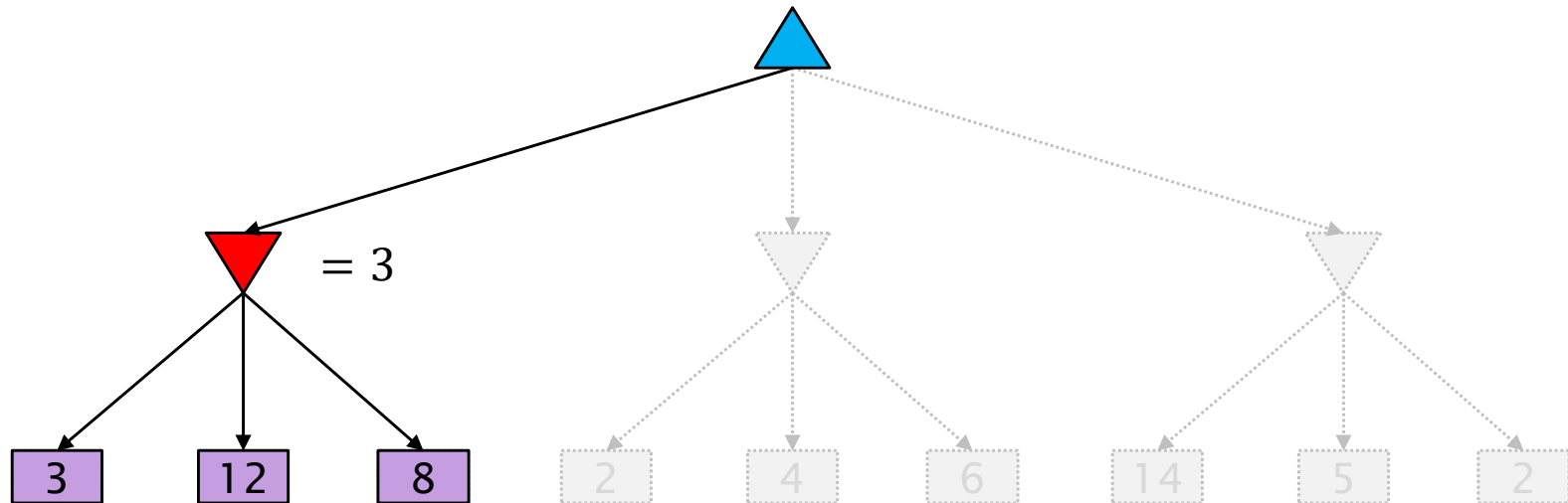
الگوریتم Minimax: یک نگاه دقیق‌تر

۵۰



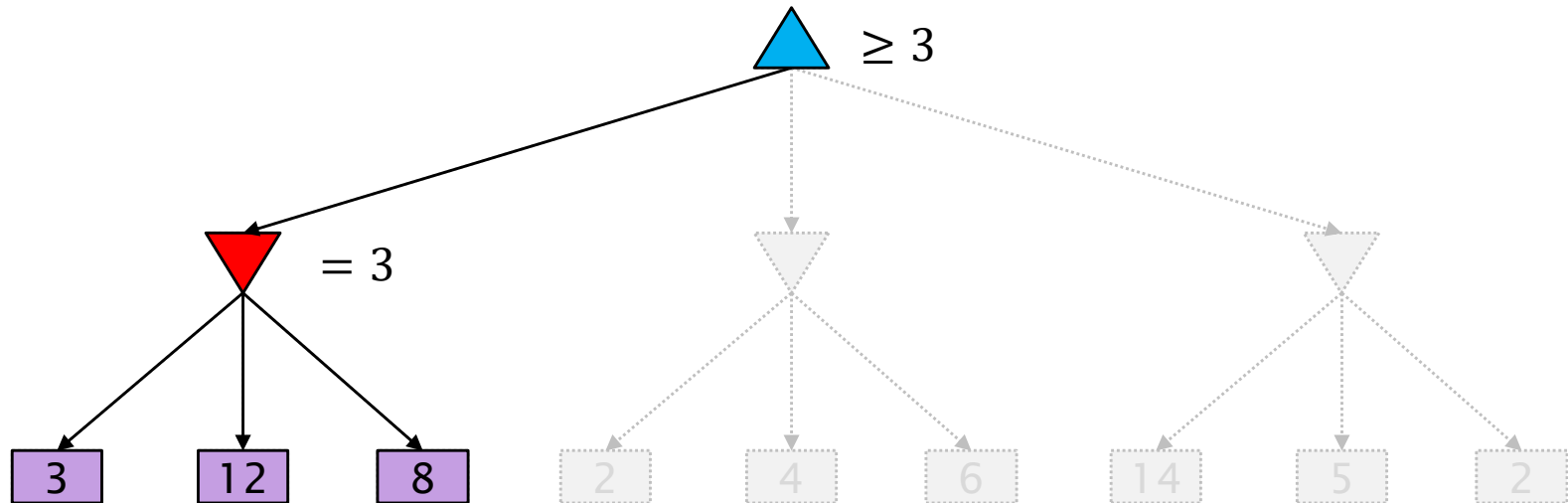
الگوریتم Minimax: یک نگاه دقیق‌تر

۵۱



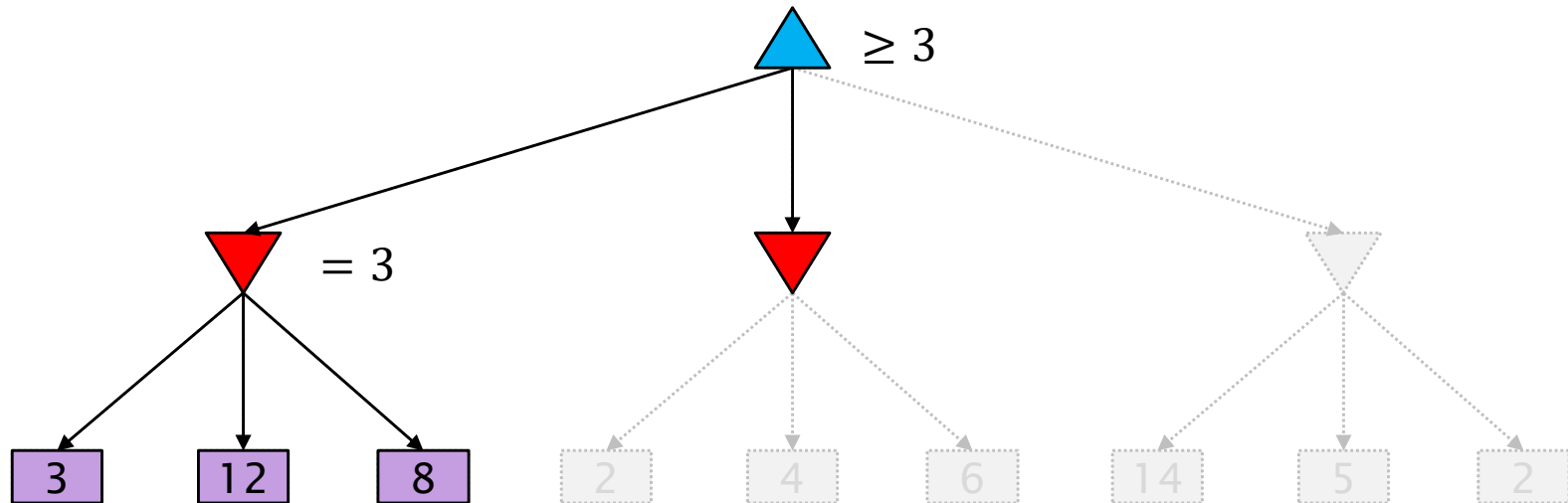
الگوریتم Minimax: یک نگاه دقیق‌تر

۵۲



الگوریتم Minimax: یک نگاه دقیق‌تر

۵۳



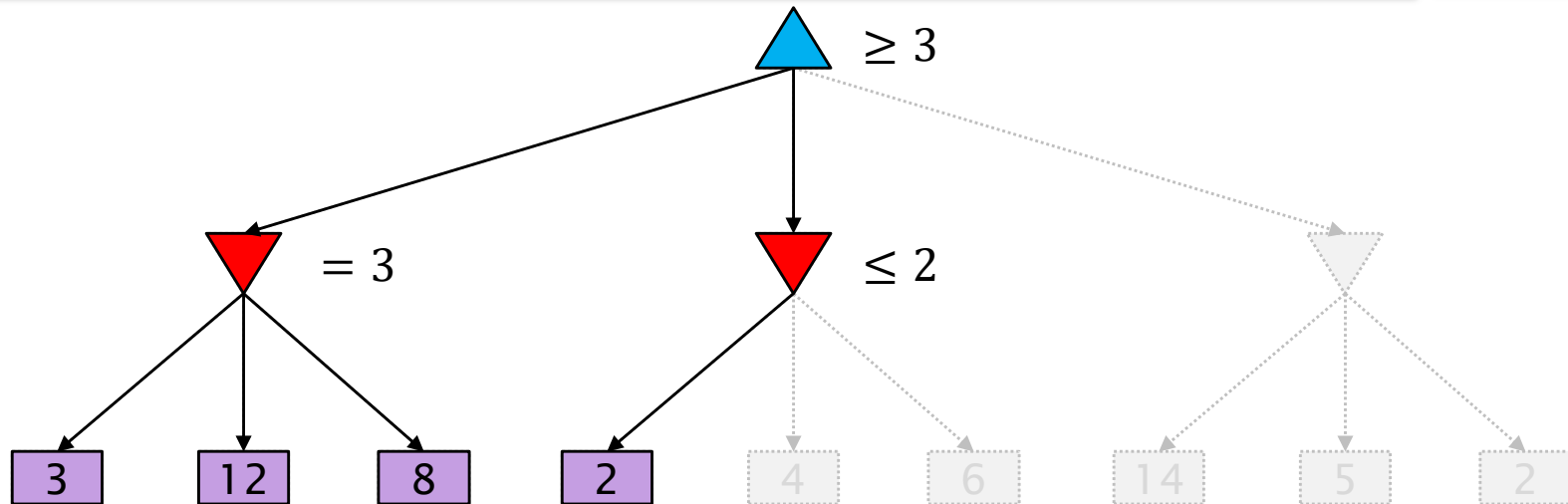
الگوریتم Minimax: یک نگاه دقیق‌تر

۵۴



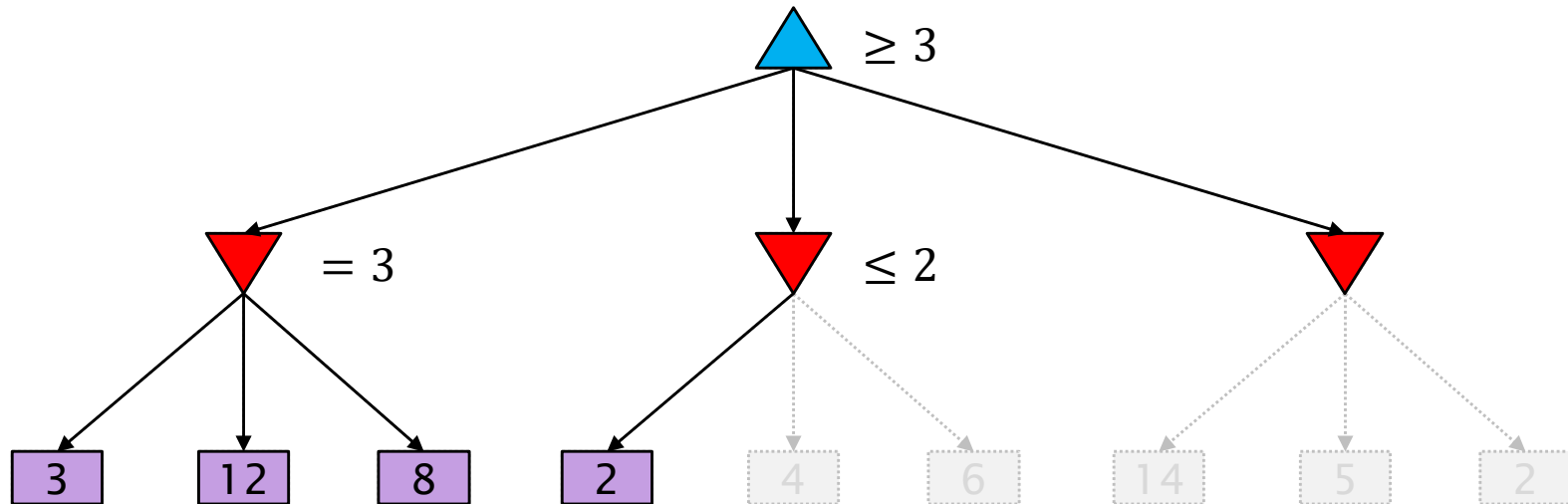
«وقتی فهمیدی چیزی بد است، دیگر برای آن که بفهمی آن چیز واقعاً چقدر بد است وقت تلف نکن.»

-- پاتریک وینستون



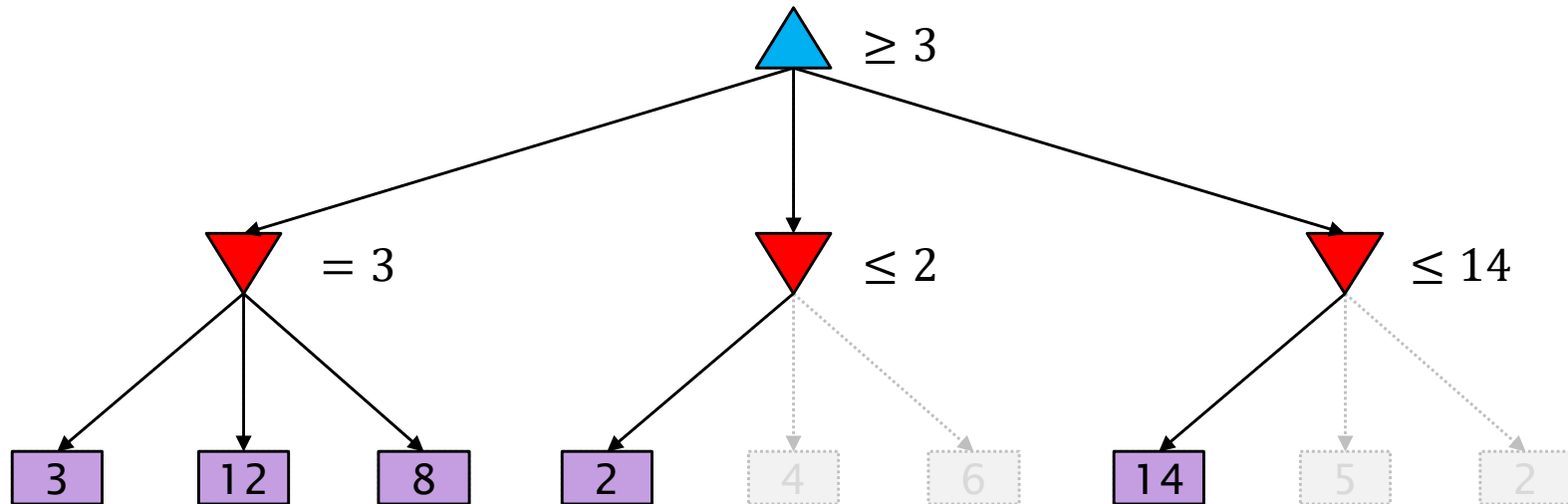
الگوریتم Minimax: یک نگاه دقیق‌تر

۵۵



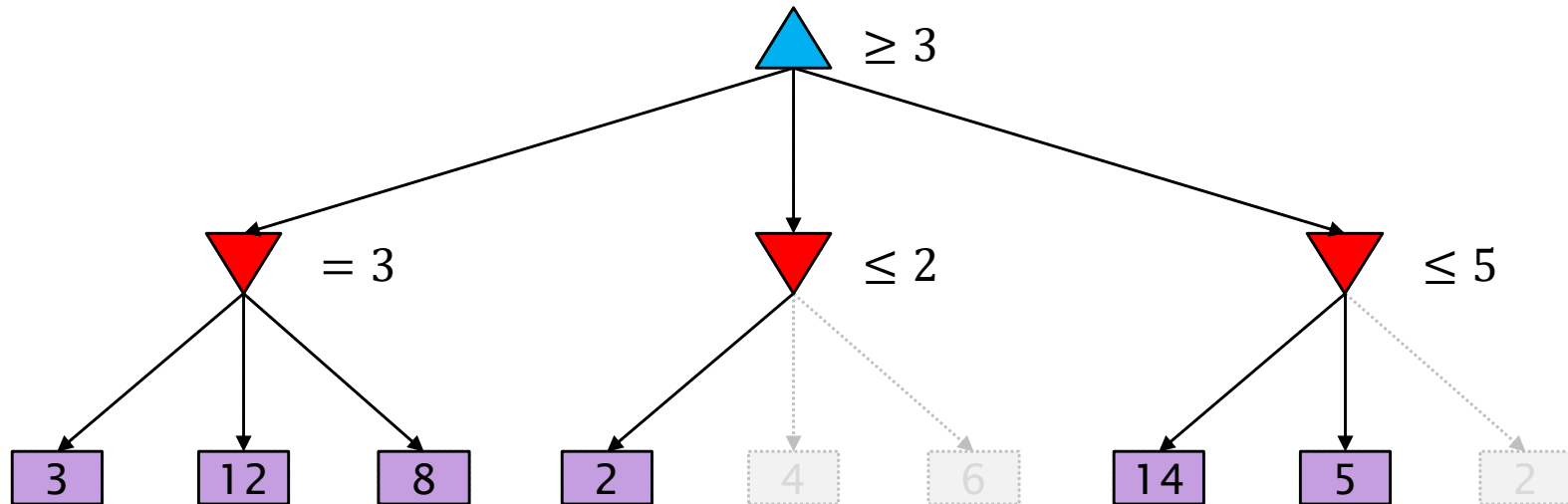
الگوریتم Minimax: یک نگاه دقیق‌تر

۵۶



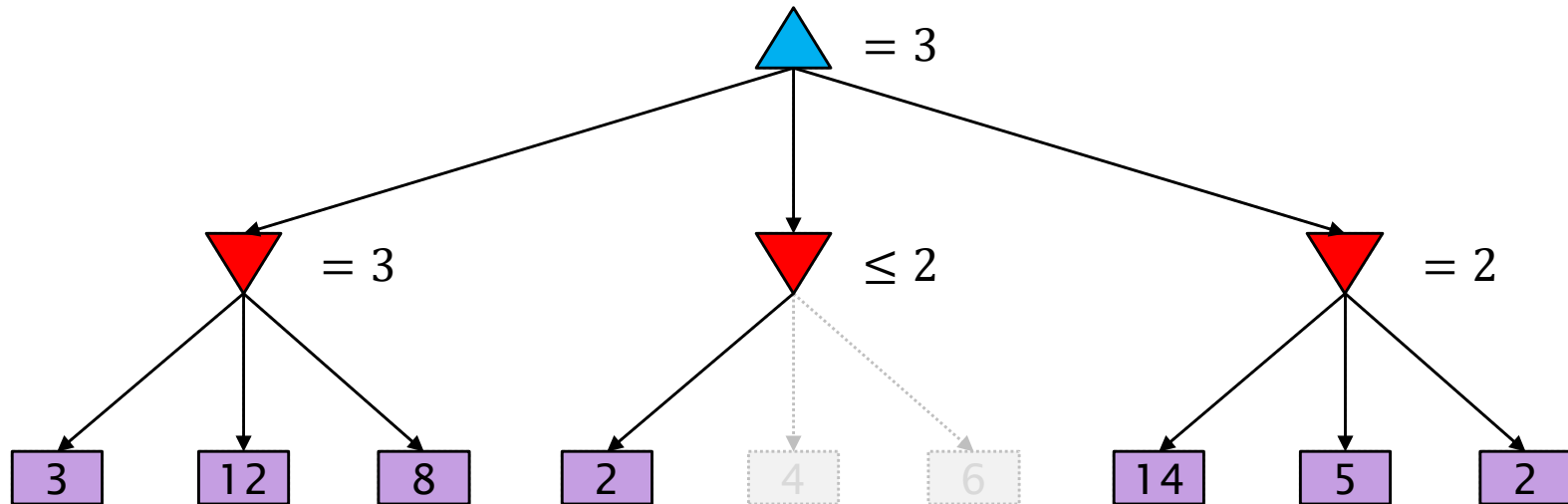
الگوریتم Minimax: یک نگاه دقیق‌تر

۵۷



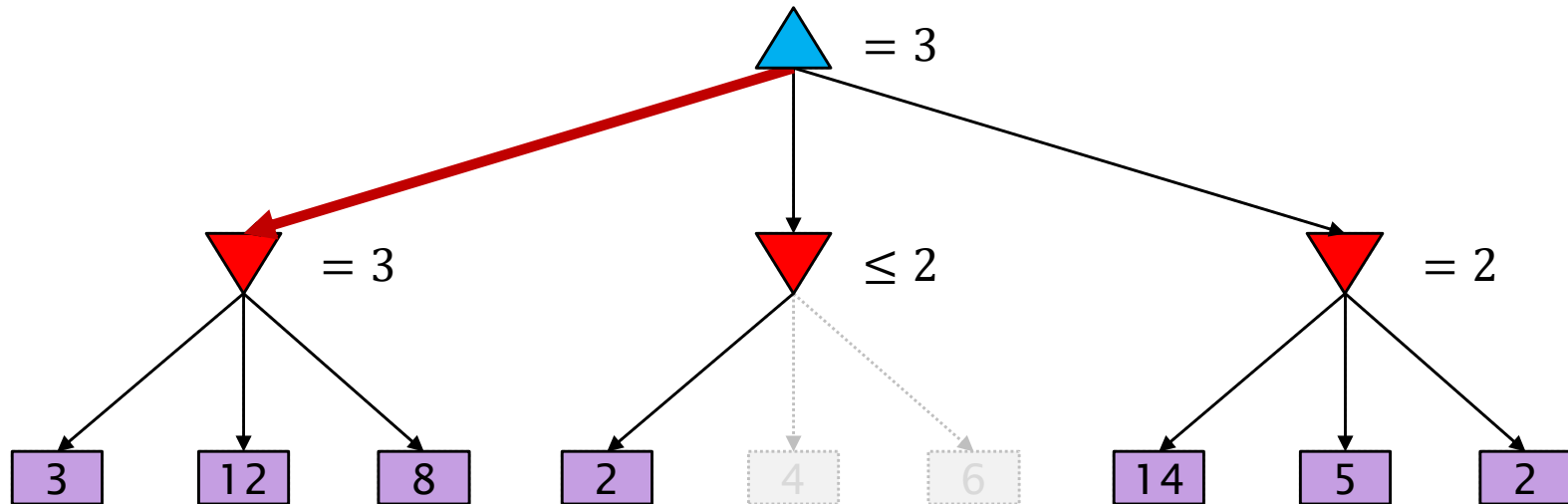
الگوریتم Minimax: یک نگاه دقیق‌تر

۵۸



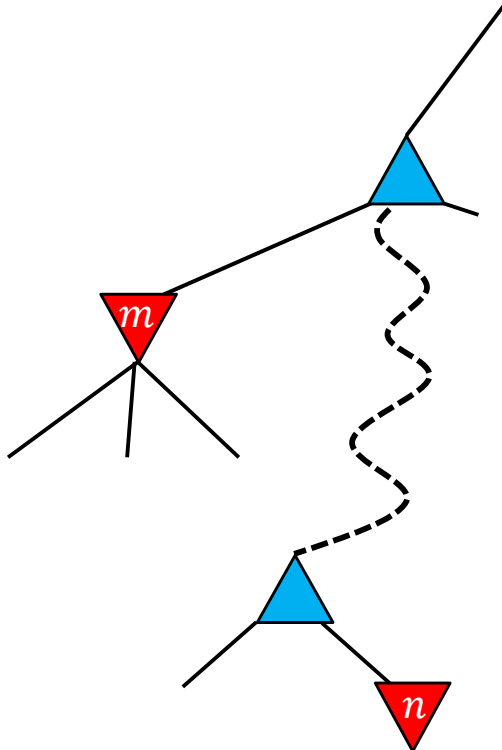
الگوریتم Minimax: یک نگاه دقیق‌تر

۵۹



هرس آلفا-بتا

۶۰



□ یک پیکره‌بندی عمومی. [نسخه MIN]

- می‌خواهیم مقدار گره n را محاسبه کنیم.
- اکنون در حال بررسی فرزندان n هستیم.
- با در نظر گرفتن فرزندان بیشتر، مقدار n تنها کاهش می‌یابد.
- مقدار n برای چه کسی مهم است؟ MAX
- فرض کنید در طول مسیر فعلی از ریشه، a بیشترین امتیازی است که MAX می‌تواند به دست آورد.
- اگر مقدار n از a کمتر شود، MAX از این گره اجتناب می‌کند و بنابراین می‌توانیم از بررسی سایر فرزندان n صرف نظر کنیم (همین الان هم مقدار n آن قدر بد هست که بازی نشود).

پیاده‌سازی هرس آلفا-بتا

۶۱

آلفا: بهترین گزینه برای بازیکن MAX روی مسیر تا ریشه
بتا: بهترین گزینه برای بازیکن MIN روی مسیر تا ریشه

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

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

ویژگی‌های هرس آلفا-بتا

۶۲

□ هرس کردن بر روی مقدار محاسبه شده تأثیری ندارد!

□ مقادیر محاسبه شده برای گره‌های میانی ممکن است نادرست باشند.

□ مهم: مقادیر محاسبه شده برای فرزندان ریشه ممکن است نادرست باشند.

□ بنابراین یک پیاده‌سازی خام به شما اجازه انتخاب صحیح اعمال را نمی‌دهد.

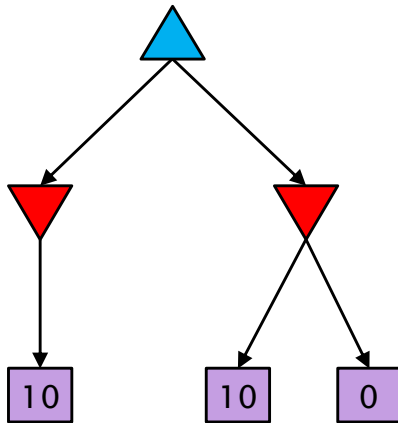
□ ترتیب فرزندان یک گره بر روی کارایی هرس کردن تأثیر می‌گذارد.

□ با یک «ترتیب عالی»:

□ پیچیدگی زمانی به $O(b^{m/2})$ کاهش می‌یابد.

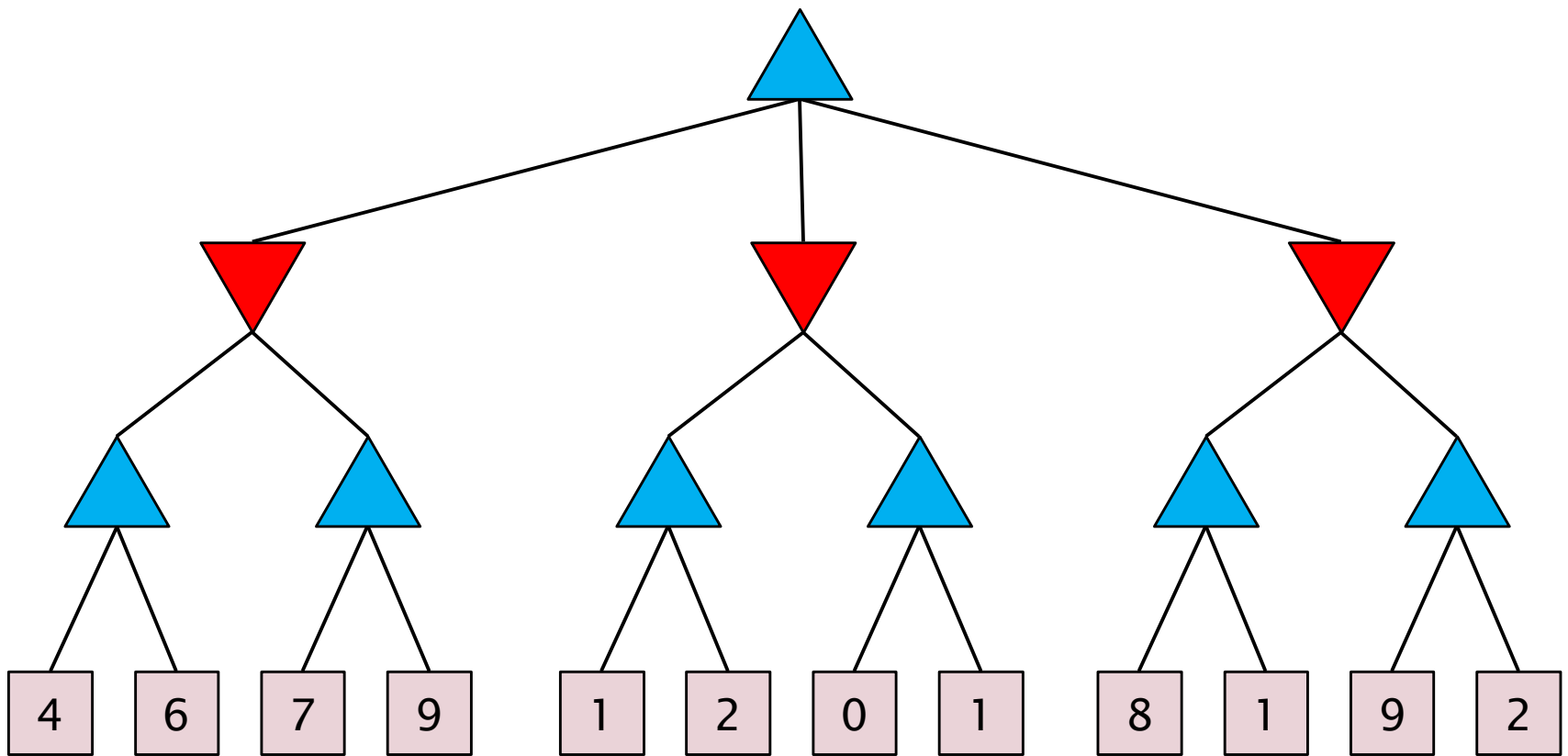
□ بنابراین عمق قابل جستجو دو برابر می‌شود.

□ با این حال هنوز هم نمی‌توان به عنوان مثال درخت شطرنج را به طور کامل تا انتها جستجو کرد.



هرس آلفا-بتا: اجرای نمایشی

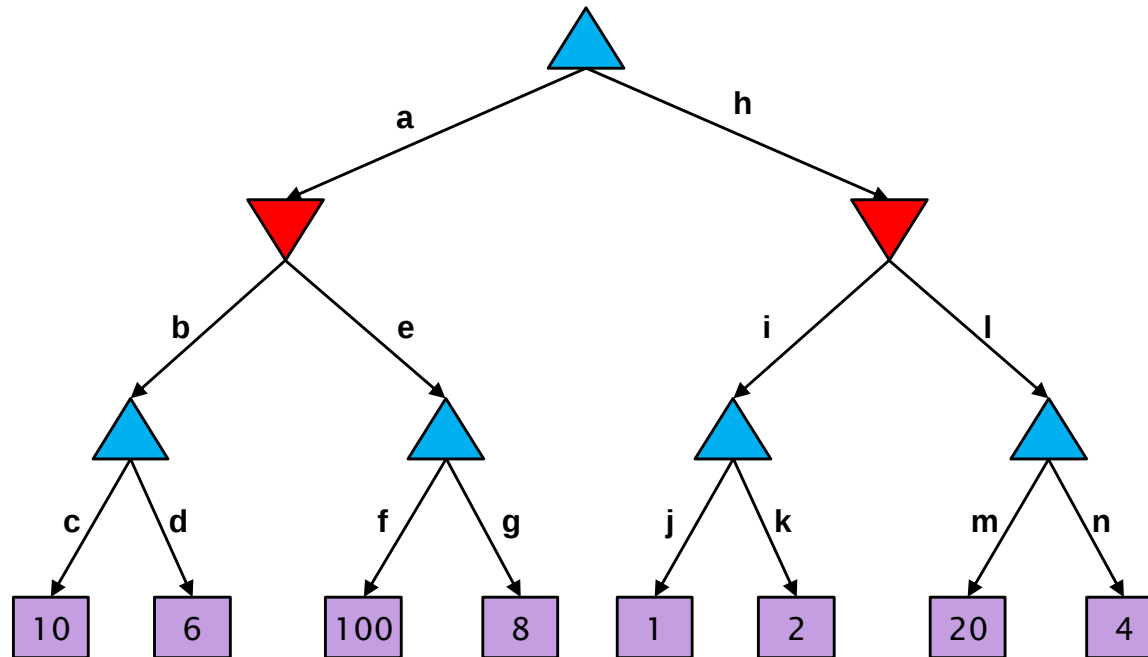
۶۳



پرسش کلاسی

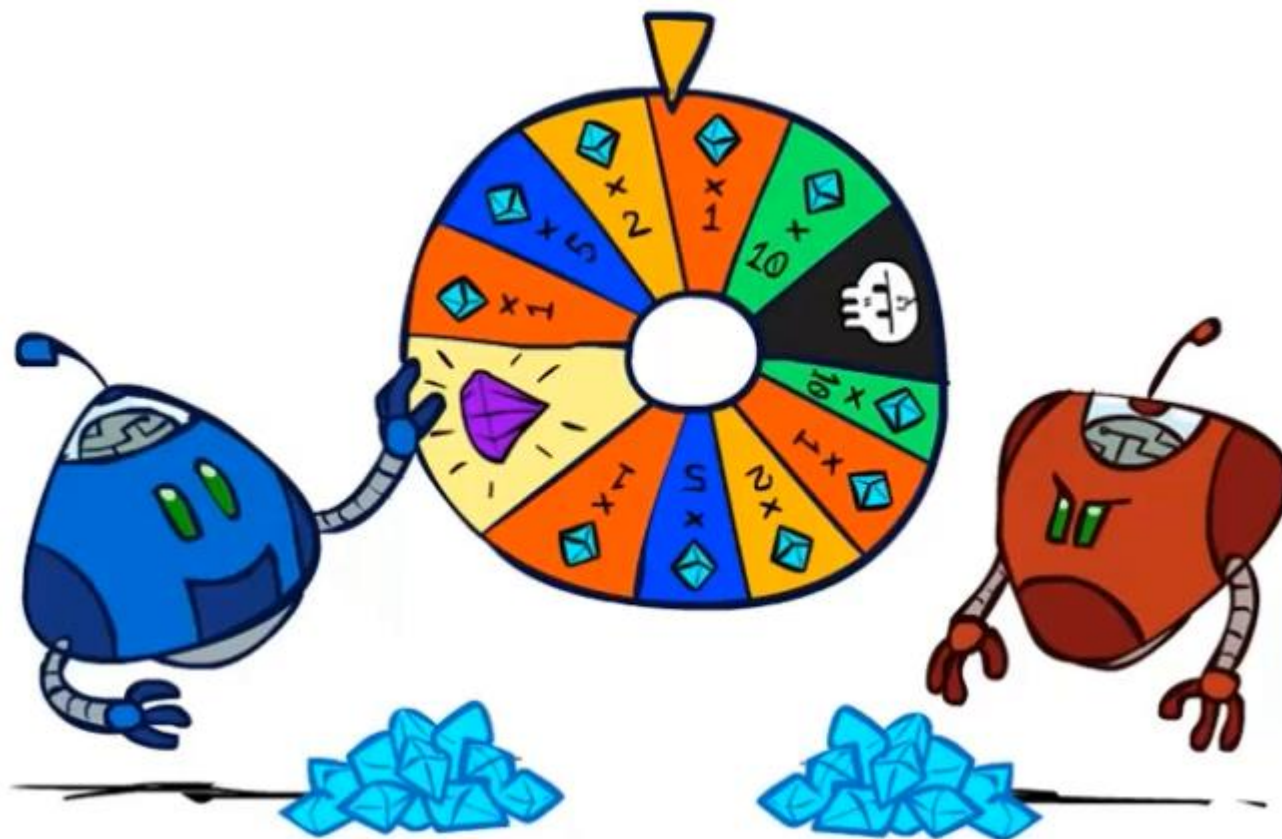
۶۴

□ کدام یک از شاخه‌های درخت زیر با استفاده از هرس آلفا-بتا هرس می‌شوند؟



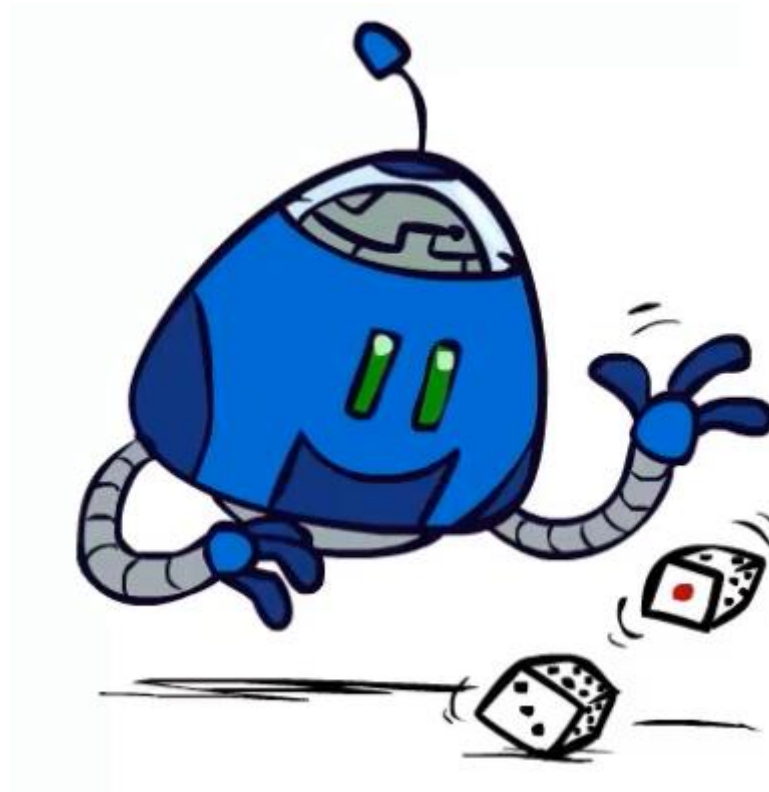
عدم قطعیت و توابع سودمندی

۶۵



نتایج غیرقطعی

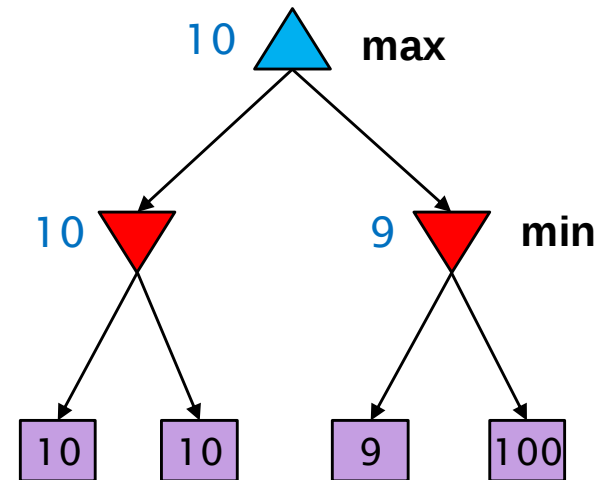
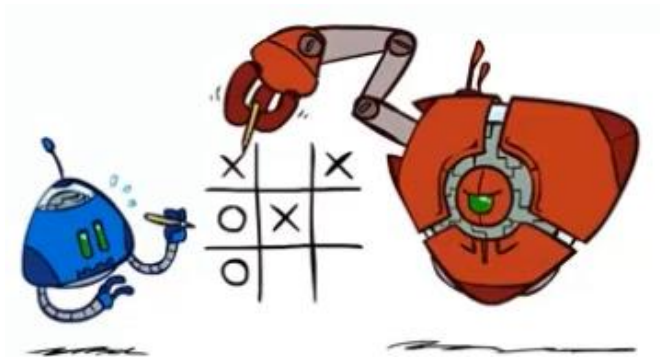
۶۶



مقایسه بدترین حالت و حالت متوسط

۶۷

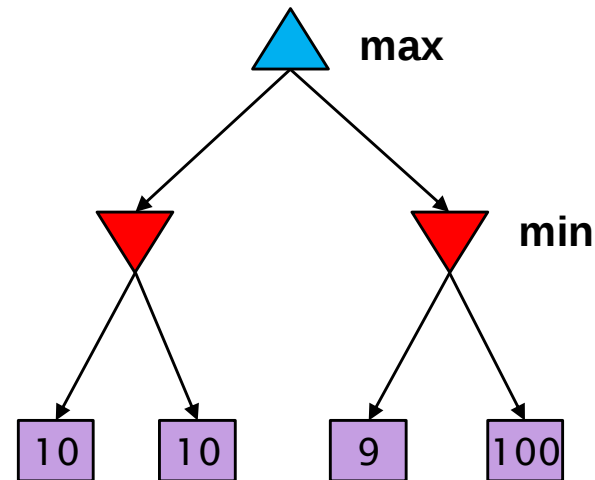
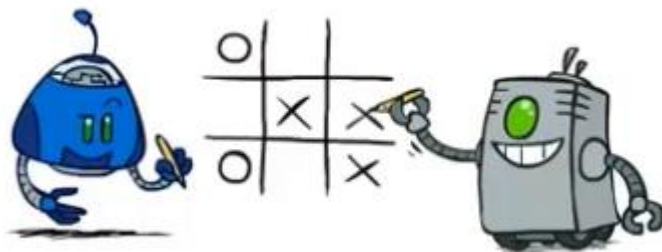
□ ایده. نتایج غیرقطعی به وسیله شانس کنترل می‌شوند، نه رقیب!



مقایسه بدترین حالت و حالت متوسط

۶۸

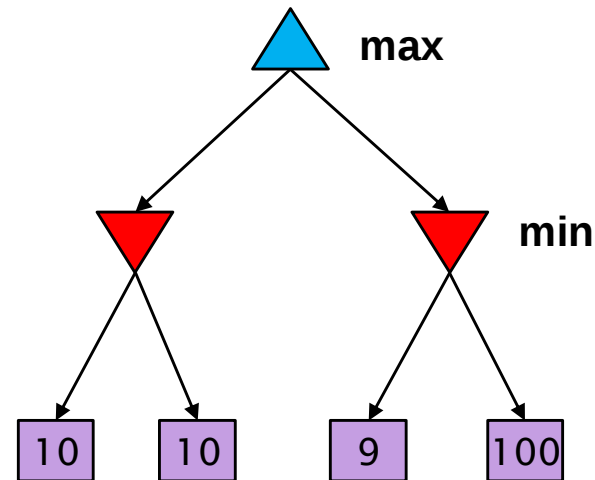
□ ایده. نتایج غیرقطعی به وسیله شانس کنترل می‌شوند، نه رقیب!



مقایسه بدترین حالت و حالت متوسط

۶۹

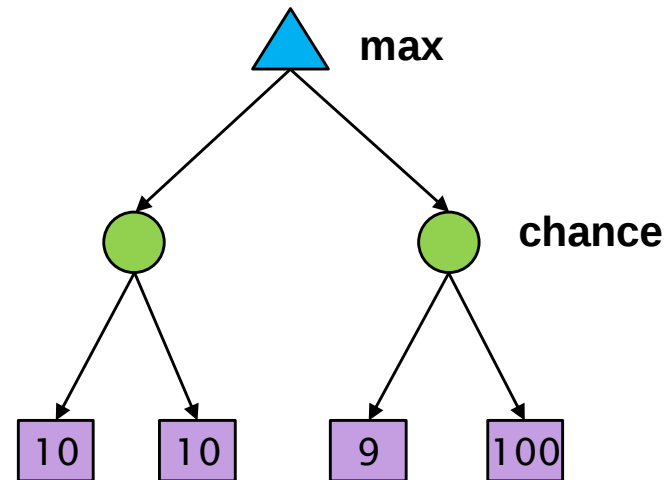
□ ایده. نتایج غیرقطعی به وسیله شانس کنترل می‌شوند، نه رقیب!



مقایسه بدترین حالت و حالت متوسط

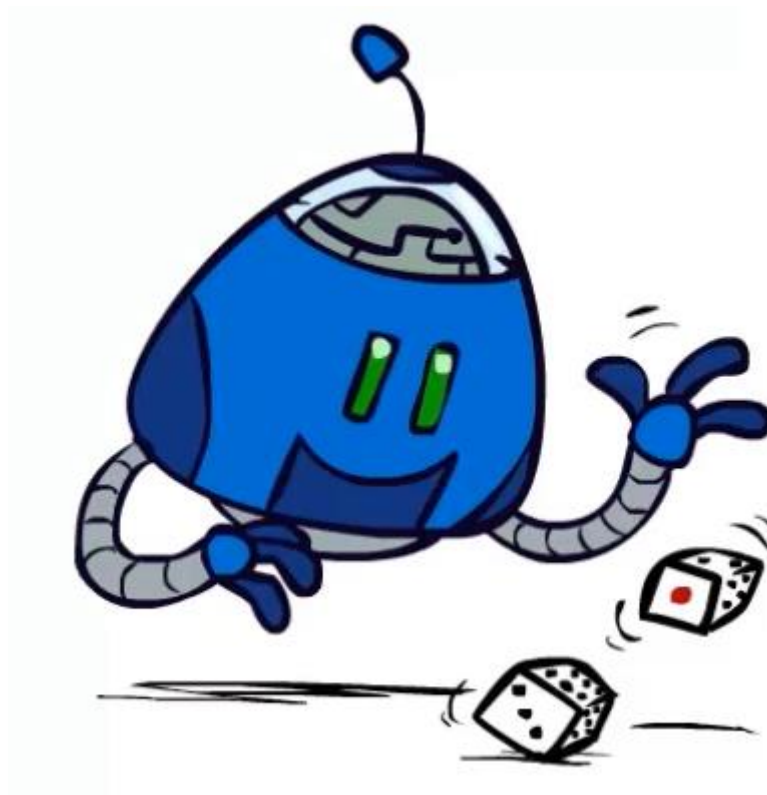
۷۰

□ ایده. نتایج غیرقطعی به وسیله شانس کنترل می‌شوند، نه رقیب!



جستجوی Expectimax

۷۱



جستجوی Expectimax

۷۲

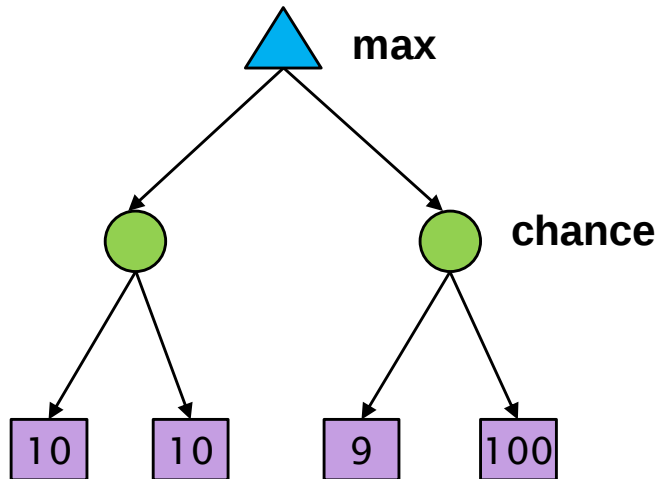
□ س. چگونه ممکن است نتیجه یک عمل را به صورت قطعی ندانیم؟

□ تصادفی بودن به طور صریح: مانند پرتاب تاس

□ رقبای غیر قابل پیش‌بینی: ارواحی که به صورت تصادفی واکنش نشان می‌دهند.

□ عملیات ناموفق: در حین حرکت یک روبات، ممکن است چرخ‌ها خراب شوند.

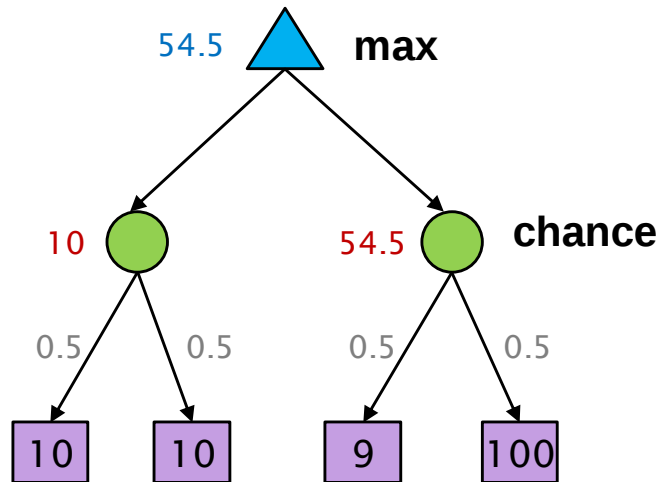
□ در این صورت مقادیر باید بیانگر حالت متوسط باشند و نه بدترین حالت.



جستجوی Expectimax

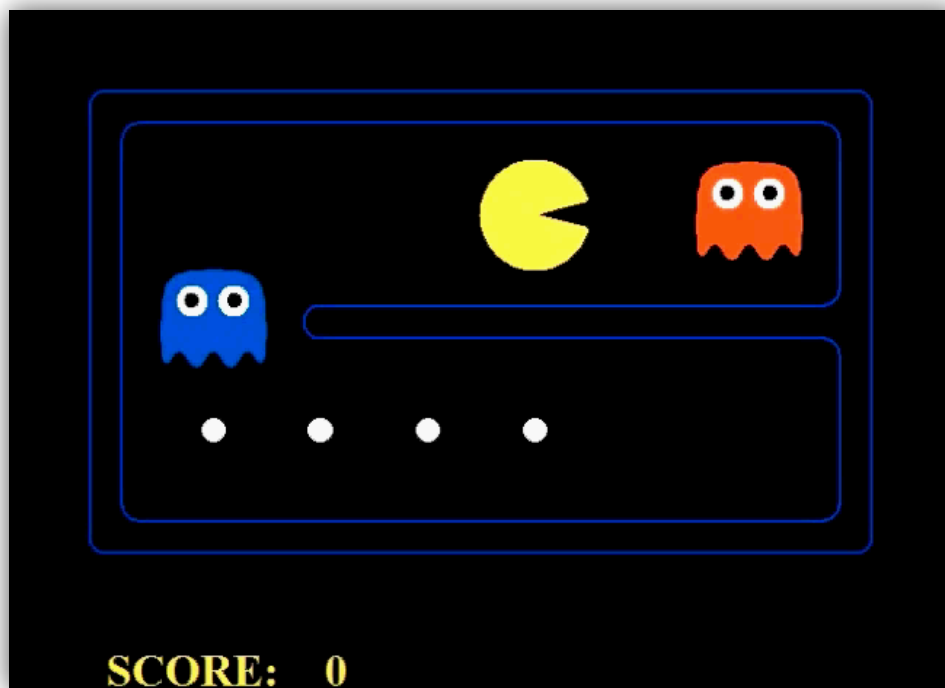
۷۳

- جستجوی Expectimax. محاسبه میانگین امتیازها تحت بازی بهینه.
- گره‌های max همانند جستجوی minimax
- گره‌های شانس مانند گره‌های min هستند با این تفاوت که نتیجه آنها غیرقطعی است.
- محاسبه سودمندی مورد انتظار در گره‌های شانس: میانگین وزن دار فرزندان



جستجوی Expectimax: اجرای نمایشی

۷۴



```
python pacman.py -p ExpectimaxAgent -l trappedClassic -a depth=3 -z 2 -n 10
```

Win Rate: 4/10 (0.40)

Record: Loss, Win, Loss, Loss, Loss, Win, Win, Loss, Loss

پیادهسازی Expectimax

۷۵

```
def value(state):
```

```
    if state is a TERMINAL:
```

```
        if the next agent is MAX:
```

```
            if the next agent is EXP:
```

```
        return the state's utility
```

```
        return max-value(state)
```

```
        return exp-value(state)
```

```
def max-value(state):
```

```
    initialize  $v = -\infty$ 
```

```
    for each successor of state:
```

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

```
    return v
```

```
def exp-value(state):
```

```
    initialize  $v = 0$ 
```

```
    for each successor of state:
```

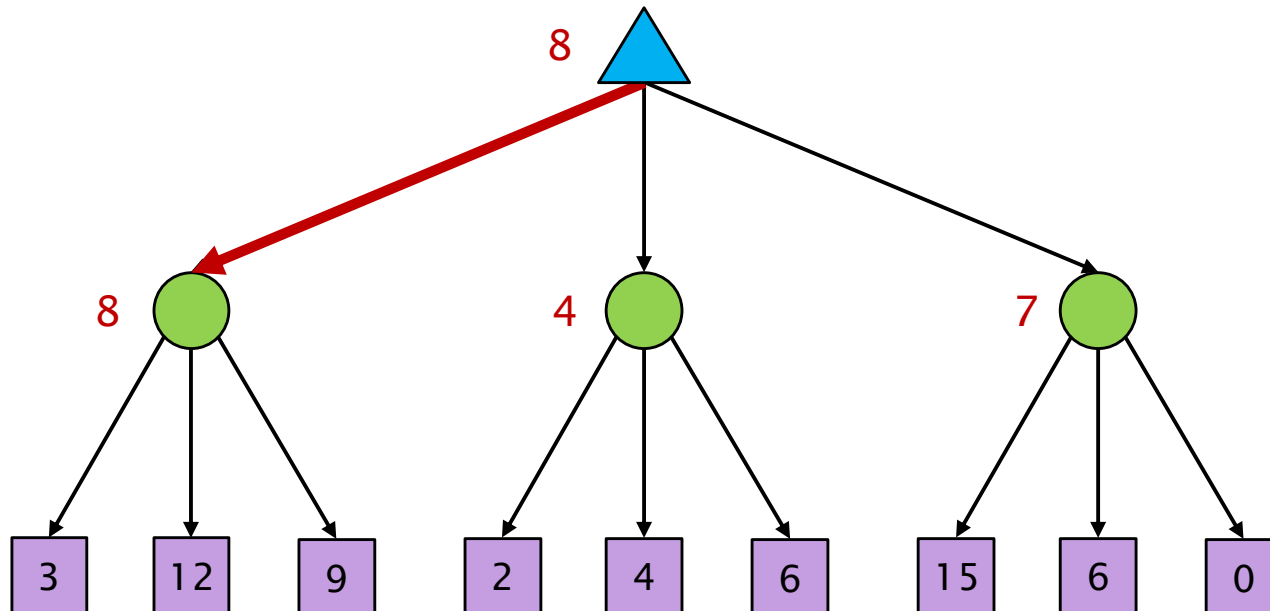
```
         $p = \text{probability}(\text{successor})$ 
```

```
         $v += p * \text{value}(\text{successor})$ 
```

```
    return v
```

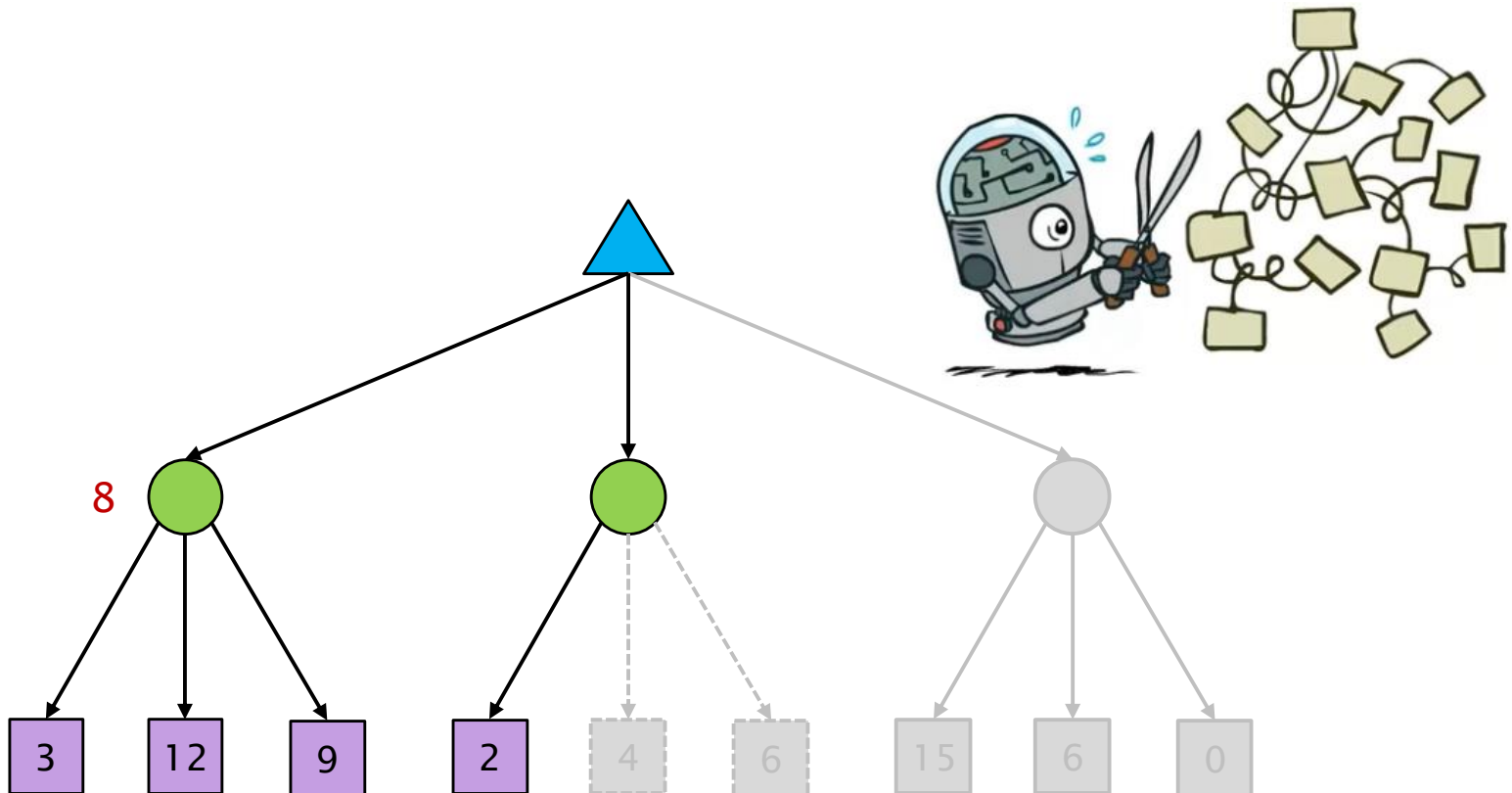
مثال Expectimax

۷۶



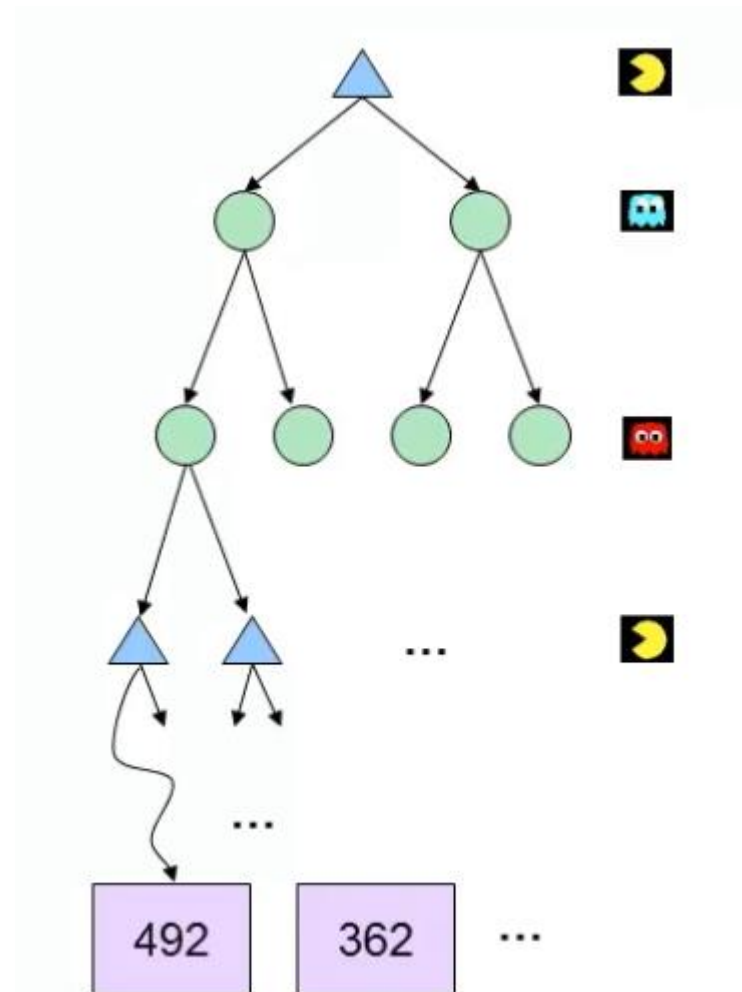
آیا هرس درخت Expectimax امکان دارد؟

۷۷



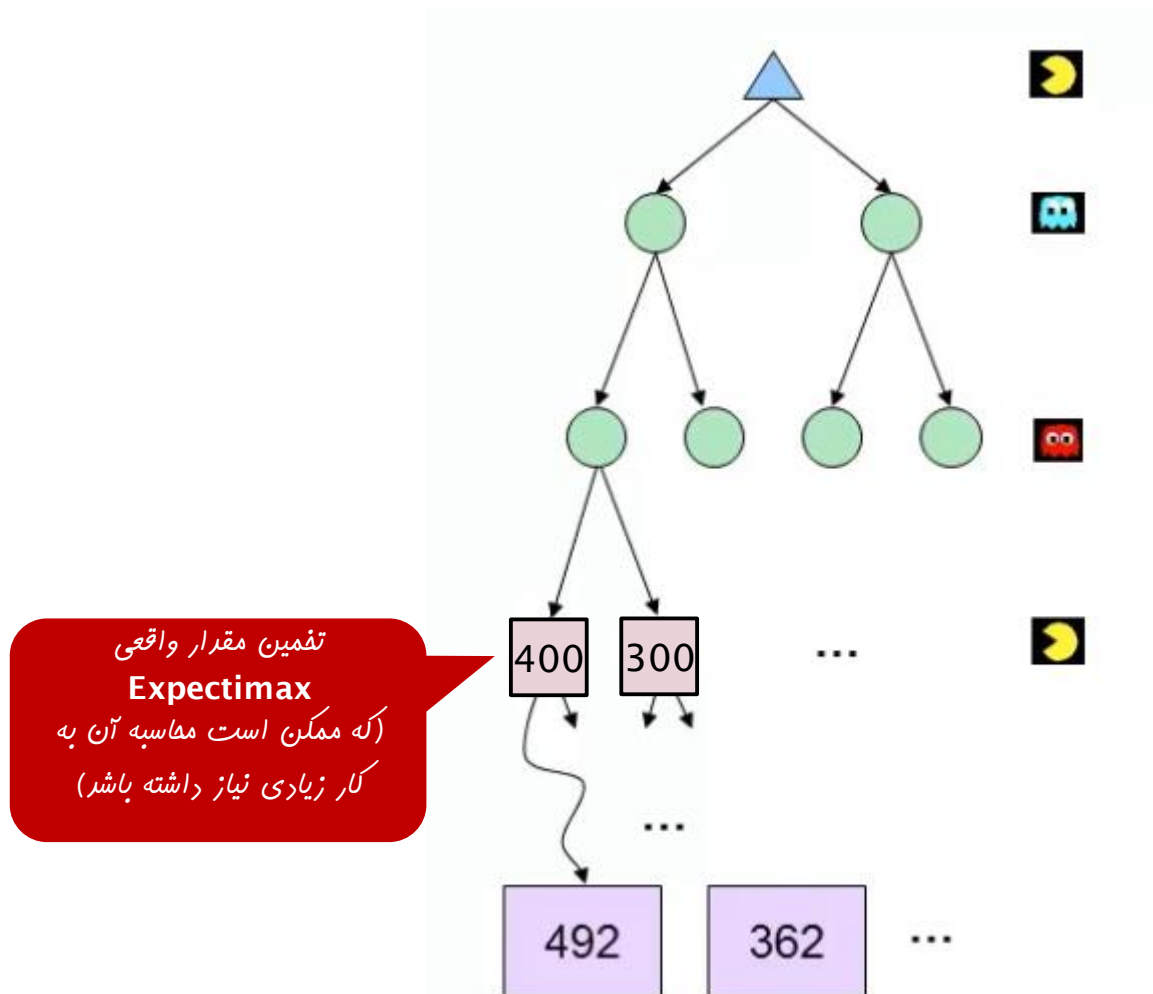
محدود کردن عمق جستجو

۷۸



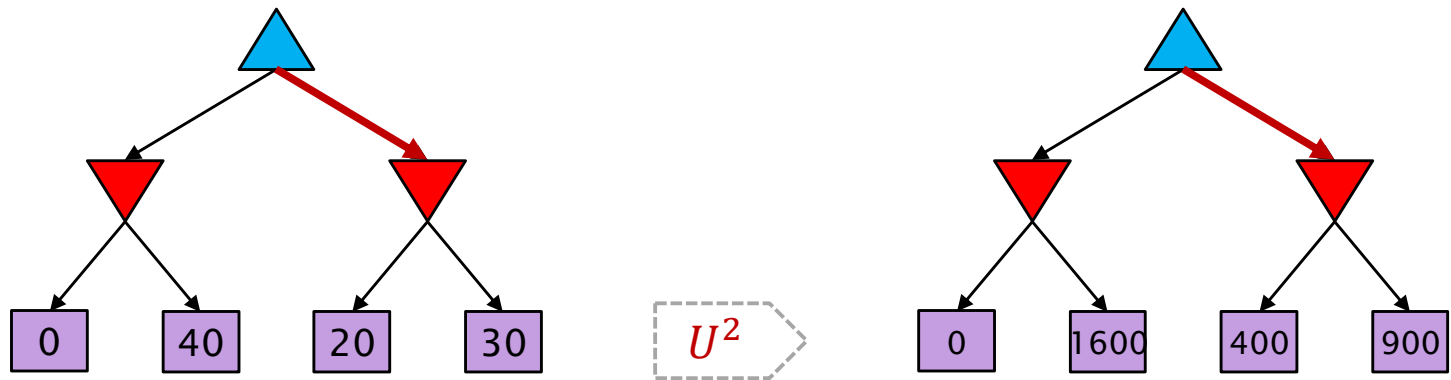
محدود کردن عمق جستجو

۷۹



مقادیر سودمندی

۸۰



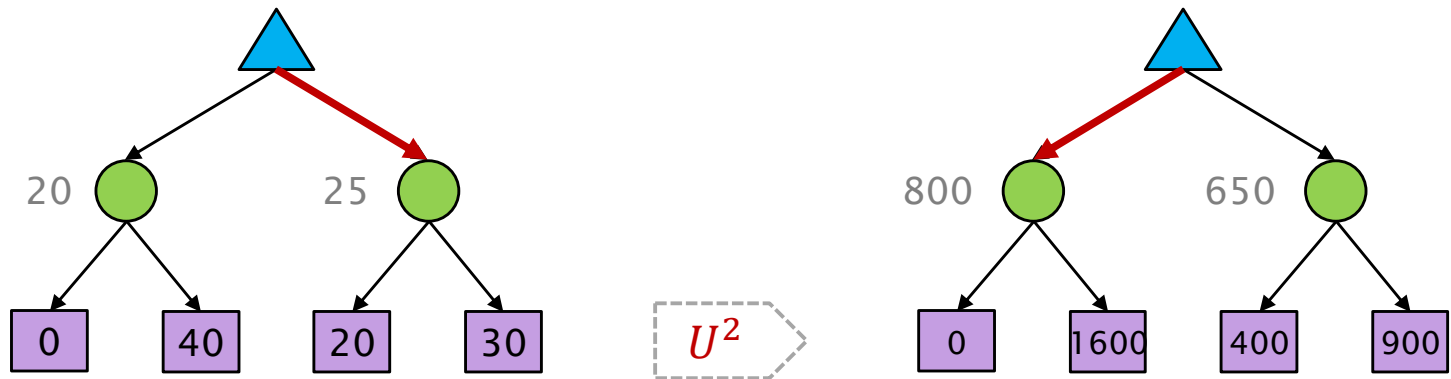
□ در جستجوی Minimax مقادیر دقیق سودمندی‌ها مهم نیستند و تنها ترتیب نسبی آنها اهمیت دارد.

□ فقط نیاز داریم حالت‌های بهتر سودمندی بالاتری داشته باشند.

□ بنابراین، هر تبدیلی که ترتیب نسبی مقادیر سودمندی را حفظ کند، مجاز است.

مقادیر سودمندی

۸۱



□ در جستجوی Minimax مقادیر دقیق سودمندی‌ها مهم نیستند و تنها ترتیب نسبی آنها اهمیت دارد.

□ فقط نیاز داریم حالت‌های بهتر سودمندی بالاتری داشته باشند.

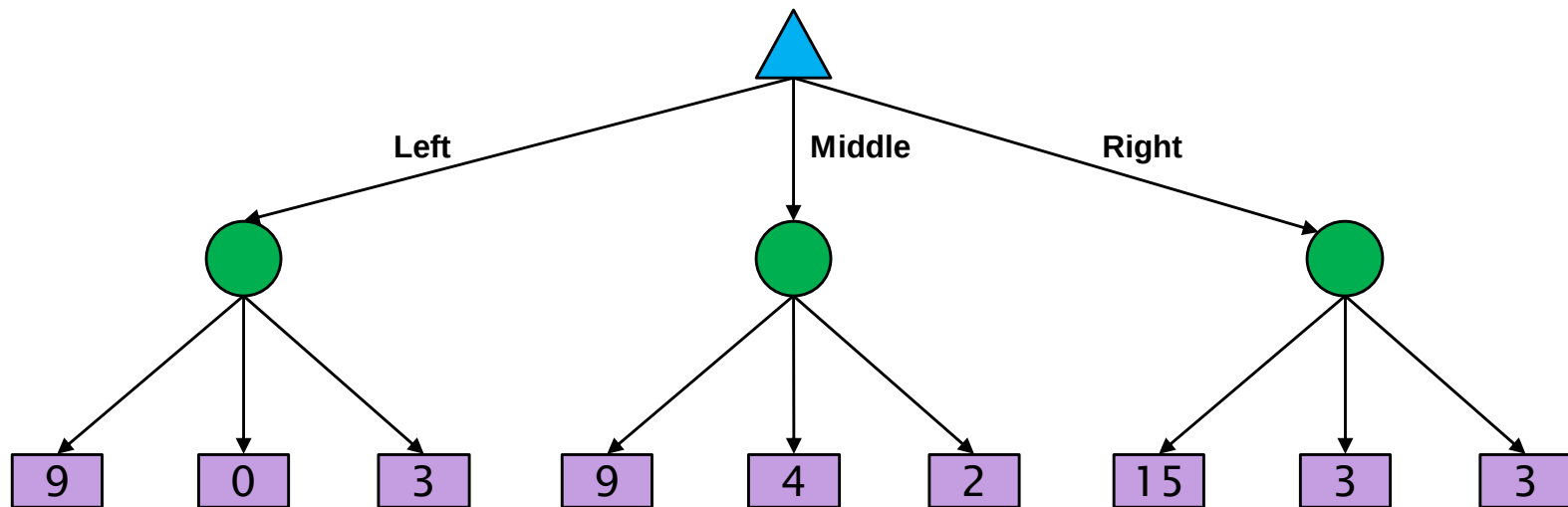
□ بنابراین، هر تبدیلی که ترتیب نسبی مقادیر سودمندی را حفظ کند، مجاز است.

□ در جستجوی Expectimax مقدار دقیق سودمندی‌ها اهمیت دارند!

پرسش کلاسی

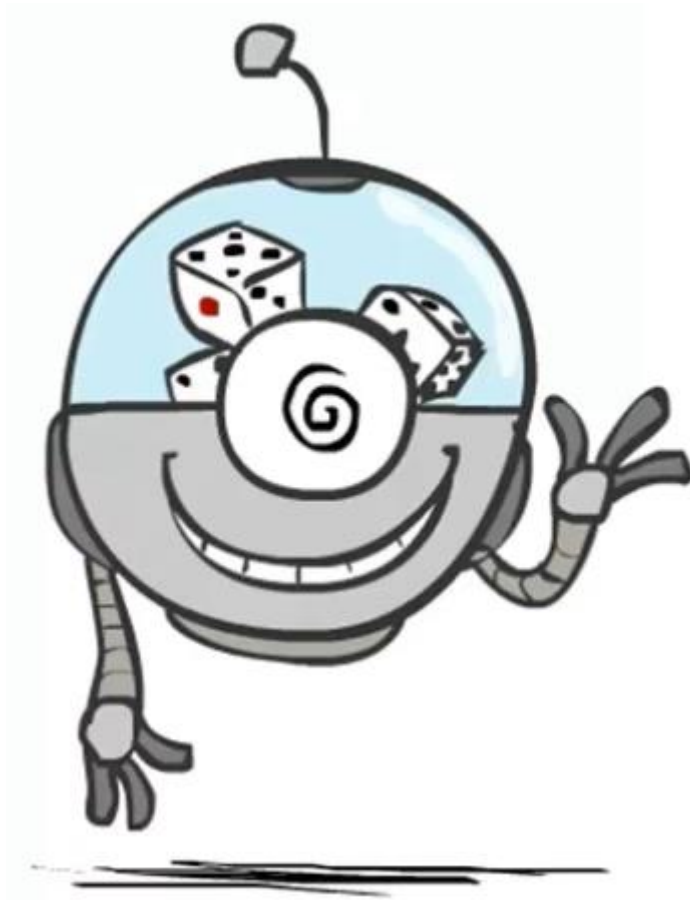
۸۲

□ درخت بازی زیر را در نظر بگیرید. با این فرض که تمام پیامدها در گره‌های شانس دارای احتمال برابر هستند، به سوالات زیر پاسخ دهید.



□ مقدار expectimax این درخت چیست؟

□ بر اساس استراتژی expectimax بازیکن بیشینه‌ساز چه عملی را انتخاب می‌کند؟



احتمالات: یادآوری

۸۴



0.25

□ متغیر تصادفی. بیانگر رویدادی که نتیجه‌اش از قبل معلوم نیست.

□ توزیع احتمال. انتساب وزن به هر یک از نتایج ممکن.



0.50

□ مثال. ترافیک در آزاد راه

□ متغیر تصادفی: T

□ نتایج ممکن: { هیچ، سبک، سنگین }

□ توزیع احتمال: { هیچ (۰/۲۵)، سبک (۰/۵۰)، سنگین (۰/۲۵) }



0.25

□ برخی از قوانین پایه‌ای در احتمالات.

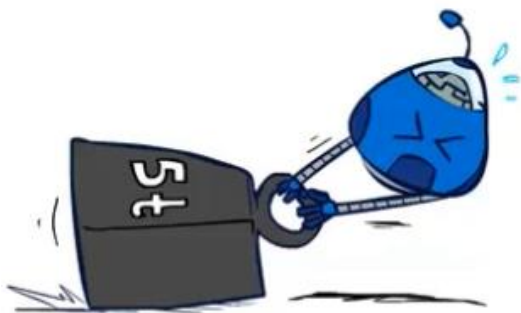
□ مقادیر احتمالات همواره غیر منفی هستند.

□ مجموع مقادیر احتمالات بر روی تمام نتایج ممکن برابر با یک است.

امید ریاضی: مقدار مورد انتظار

۸۵

□ مقدار مورد انتظار یک تابع از یک متغیر تصادفی برابر است با میانگین وزنی مقادیر احتمالات بر روی تمام نتایج ممکن.



□ مثال. زمان لازم برای رسیدن به فرودگاه؟

$$0.25 \times 20 + 0.50 \times 50 + 0.25 \times 90 = 55$$

۲۰ دقیقه



0.25

۵۰ دقیقه



0.50

۹۰ دقیقه



0.25

مقادیر احتمالات

۸۶

□ در expectimax یک مدل احتمالاتی از چگونگی عمل کردن رقیب (یا محیط) در هر حالت ممکن داریم.

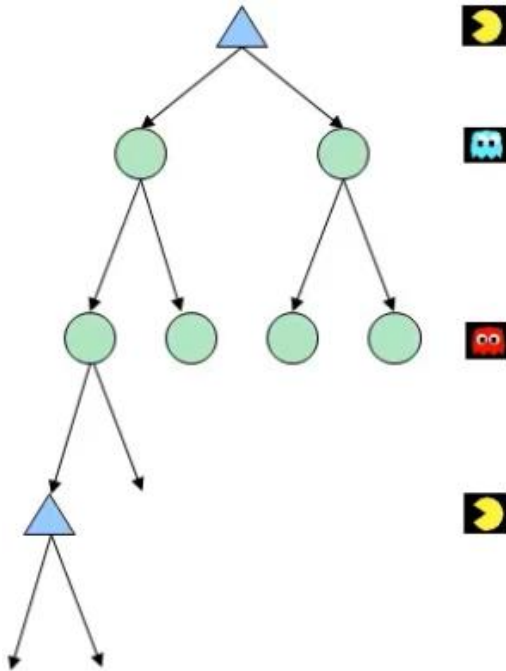
□ مدل می‌تواند یک توزیع یکنواخت ساده باشد (پرتاب تاس)

□ مدل می‌تواند بسیار پیچیده باشد و به محاسبات زیادی نیاز داشته باشد.

□ به ازای هر نتیجه‌ای که خارج از کنترل ما است، یک گره شانس داریم.

□ ممکن است مدل بگوید اعمال رقیب دارای احتمال برابر هستند.

□ فعلاً برای سادگی فرض می‌کنیم در هر یک از گره‌های شانس احتمال وقوع هر یک از فرزندان را از قبل می‌دانیم.



صرف داشتن یک باور احتمالاتی در مورد اعمال یک عامل دیگر، لزوماً به این معنا نیست که عامل دیگر به صورت تصادفی عمل می‌کند!

مدل سازی فرضیات

۸۷



خطرات خوش بینی و بدبینی

۸۸

بدبینی خطرناک

فرض بدترین حالت وقتی که این گونه نیست



خوش بینی خطرناک

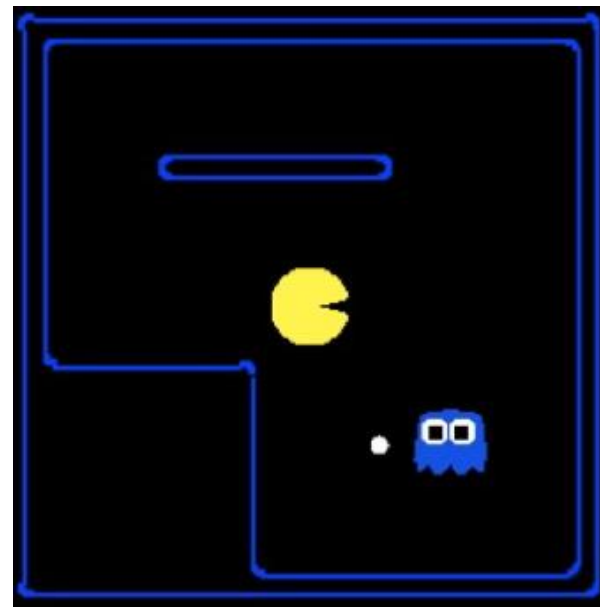
تکیه بر شانس وقتی که محیط خصمانه است



فرضیات و واقعیت‌ها

۸۹

روح با رفتار تصادفی	روح با رفتار خصمانه	
برد ۵/۵ میانگین امتیاز: ۴۹۳	برد ۵/۵ میانگین امتیاز: ۴۸۳	پکمن minimax
برد ۵/۵ میانگین امتیاز: ۵۰۳	برد ۱/۵ میانگین امتیاز: -۳۰۳	پکمن expectimax



نتایج حاصل از ۵ بار بازی

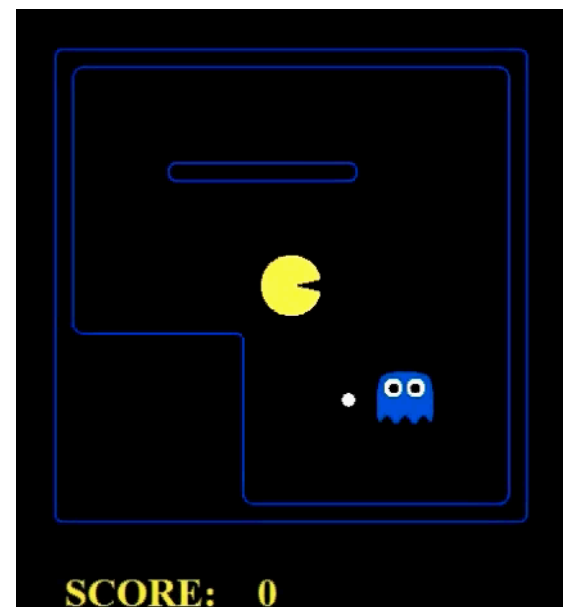
- پکمن تا عمق ۴ جستجو می‌کند و تابع ارزیابی آن به گونه‌ای است که باعث می‌شود پکمن از دردرس دوری کند.
- روح تا عمق ۲ جستجو می‌کند و تابع ارزیابی آن به گونه‌ای است که باعث می‌شود به دنبال پکمن بگردد.

فرضیات و واقعیت‌ها

۹۰

روح با رفتار تصادفی	روح با رفتار خصمانه	
برد ۵/۵ میانگین امتیاز: ۴۹۳	برد ۵/۵ میانگین امتیاز: ۴۸۳	پکمن minimax
برد ۵/۵ میانگین امتیاز: ۵۰۳	برد ۱/۵ میانگین امتیاز: -۳۰۳	پکمن expectimax

نتایج حاصل از ۵ بار بازی



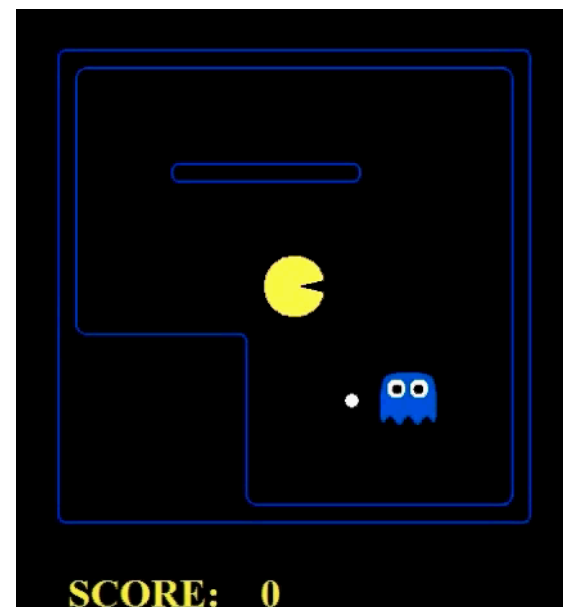
- پکمن تا عمق ۴ جستجو می‌کند و تابع ارزیابی آن به گونه‌ای است که باعث می‌شود پکمن از دردرس دوری کند.
- روح تا عمق ۲ جستجو می‌کند و تابع ارزیابی آن به گونه‌ای است که باعث می‌شود به دنبال پکمن بگردد.

فرضیات و واقعیت‌ها

۹۱

روح با رفتار تصادفی	روح با رفتار خصمانه	
برد ۵/۵ میانگین امتیاز: ۴۹۳	برد ۵/۵ میانگین امتیاز: ۴۸۳	پکمن minimax
برد ۵/۵ میانگین امتیاز: ۵۰۳	برد ۱/۵ میانگین امتیاز: -۳۰۳	پکمن expectimax

نتایج حاصل از ۵ بار بازی



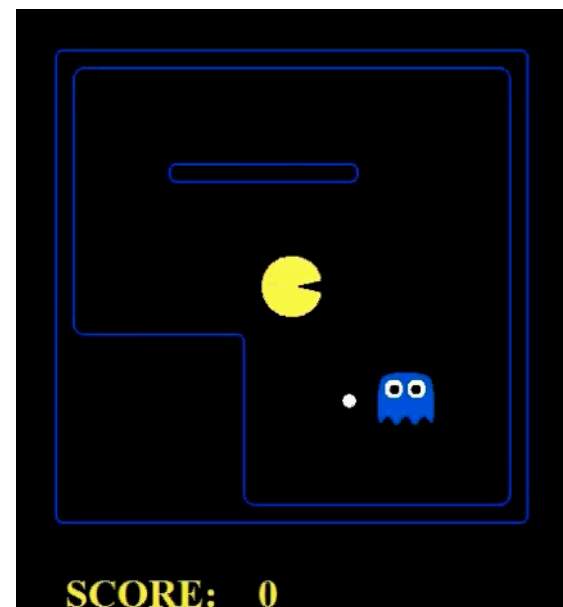
- پکمن تا عمق ۴ جستجو می‌کند و تابع ارزیابی آن به گونه‌ای است که باعث می‌شود پکمن از دردرس دوری کند.
- روح تا عمق ۲ جستجو می‌کند و تابع ارزیابی آن به گونه‌ای است که باعث می‌شود به دنبال پکمن بگردد.

فرضیات و واقعیت‌ها

۹۲

روح با رفتار تصادفی	روح با رفتار خصمانه	
برد ۵/۵ میانگین امتیاز: ۴۹۳	برد ۵/۵ میانگین امتیاز: ۴۸۳	پکمن minimax
برد ۵/۵ میانگین امتیاز: ۵۰۳	برد ۱/۵ میانگین امتیاز: -۳۰۳	پکمن expectimax

نتایج حاصل از ۵ بار بازی



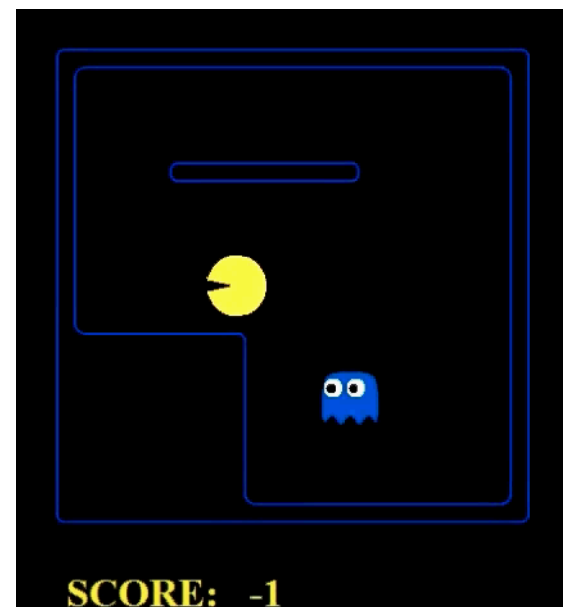
- پکمن تا عمق ۴ جستجو می‌کند و تابع ارزیابی آن به گونه‌ای است که باعث می‌شود پکمن از دردرس دوری کند.
- روح تا عمق ۲ جستجو می‌کند و تابع ارزیابی آن به گونه‌ای است که باعث می‌شود به دنبال پکمن بگردد.

فرضیات و واقعیت‌ها

۹۳

روح با رفتار تصادفی	روح با رفتار خصمانه	
برد ۵/۵ میانگین امتیاز: ۴۹۳	برد ۵/۵ میانگین امتیاز: ۴۸۳	پکمن minimax
برد ۵/۵ میانگین امتیاز: ۵۰۳	برد ۱/۵ میانگین امتیاز: -۳۰۳	پکمن expectimax

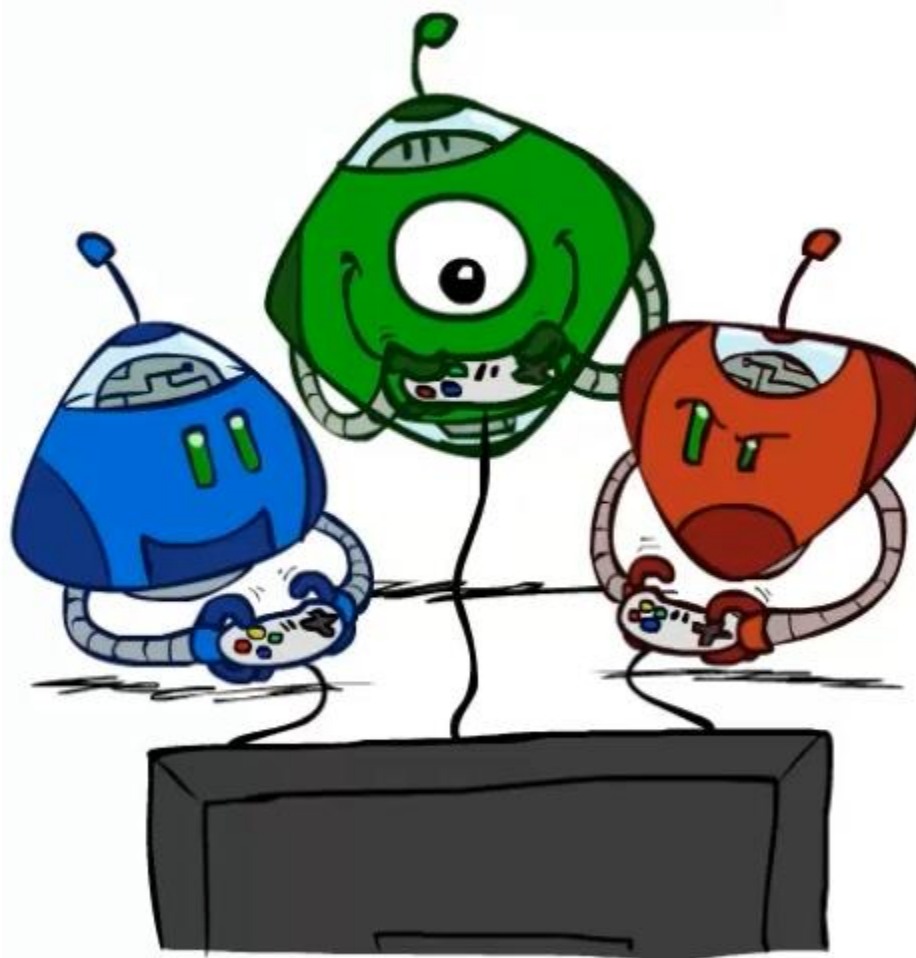
نتایج حاصل از ۵ بار بازی



- پکمن تا عمق ۴ جستجو می‌کند و تابع ارزیابی آن به گونه‌ای است که باعث می‌شود پکمن از دردرس دوری کند.
- روح تا عمق ۲ جستجو می‌کند و تابع ارزیابی آن به گونه‌ای است که باعث می‌شود به دنبال پکمن بگردد.

انواع دیگر بازی‌ها

۹۴



لایه‌های مخلوط: جستجوی Expectiminimax

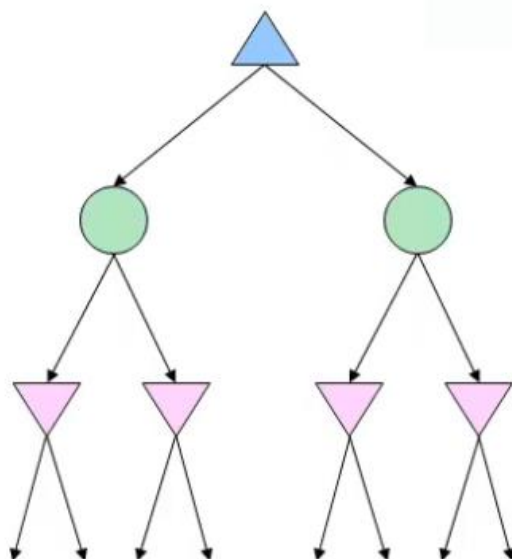
۹۵

□ مثال. بازی تخته نرد [بزرگمهر، وزیر ساسانیان]

□ جستجوی expectiminimax.

□ محیط مانند یک «بازیکن تصادفی» است که بعد از هر بازیکن حرکت می‌کند.

□ هر گره ترکیب مناسبی از مقادیر فرزندانش را محاسبه می‌کند.



داستان ساختن تخته‌نرد به روایت فردوسی

۹۶



تندیس فردوسی، رم، ایتالیا

سپاس از خداوند خورشید و ماه
چون این کار دلگیرت آمد به بُن
چنین آلهی یافت شاه جهان
که آمد فرستاده‌ی شاه هند
بیاورد پس نامه‌ای بر پرند
بدانند هر مهره‌ای را به نام
پیاده بدانند و پیل و سپاه
گر این نغزبازی بجای آورند
همان باژ و ساوی که فرمود شاه
و گر نامداران ایران گروه
چو با دانش ما ندارند تاو

که رستم ز بوزرجمهر و ز شاه
ز شطرنج باید که رانم سَفَن
ز گفتار بیدار کار آلهان
ابابیل و پتر و سواران سند
نَبشته به نوشین روان رای هند
که گویند پس فانه او کدام
رخ و اسپ و رختار خرزین و شاه
درین کار پاکیزه رای آورند
بفوبی فرستم بران بارگاه
ازین دانش آیند یکسر ستوه
نفوهند زین بوم و بر باژ و ساو ...

مثال: تخته‌نرد

۹۷



□ پرتاب تاس فاکتور انشعاب را افزایش می‌دهد:

□ دو تاس: ۲۱ حالت ممکن

□ تعداد حرکتهای قانونی: ۲۰ حرکت (تقریباً)

□ تعداد گره‌ها در عمق ۲: $20 \times (21 \times 20)^3 = 1/2 \times 10^9$

□ با افزایش عمق، احتمال رسیدن به یک گره خاص کاهش می‌یابد:

□ بنابراین میزان فایده جستجو کاهش می‌یابد.

□ در نتیجه، محدود کردن عمق آسیب چندانی به نتیجه وارد نمی‌کند.

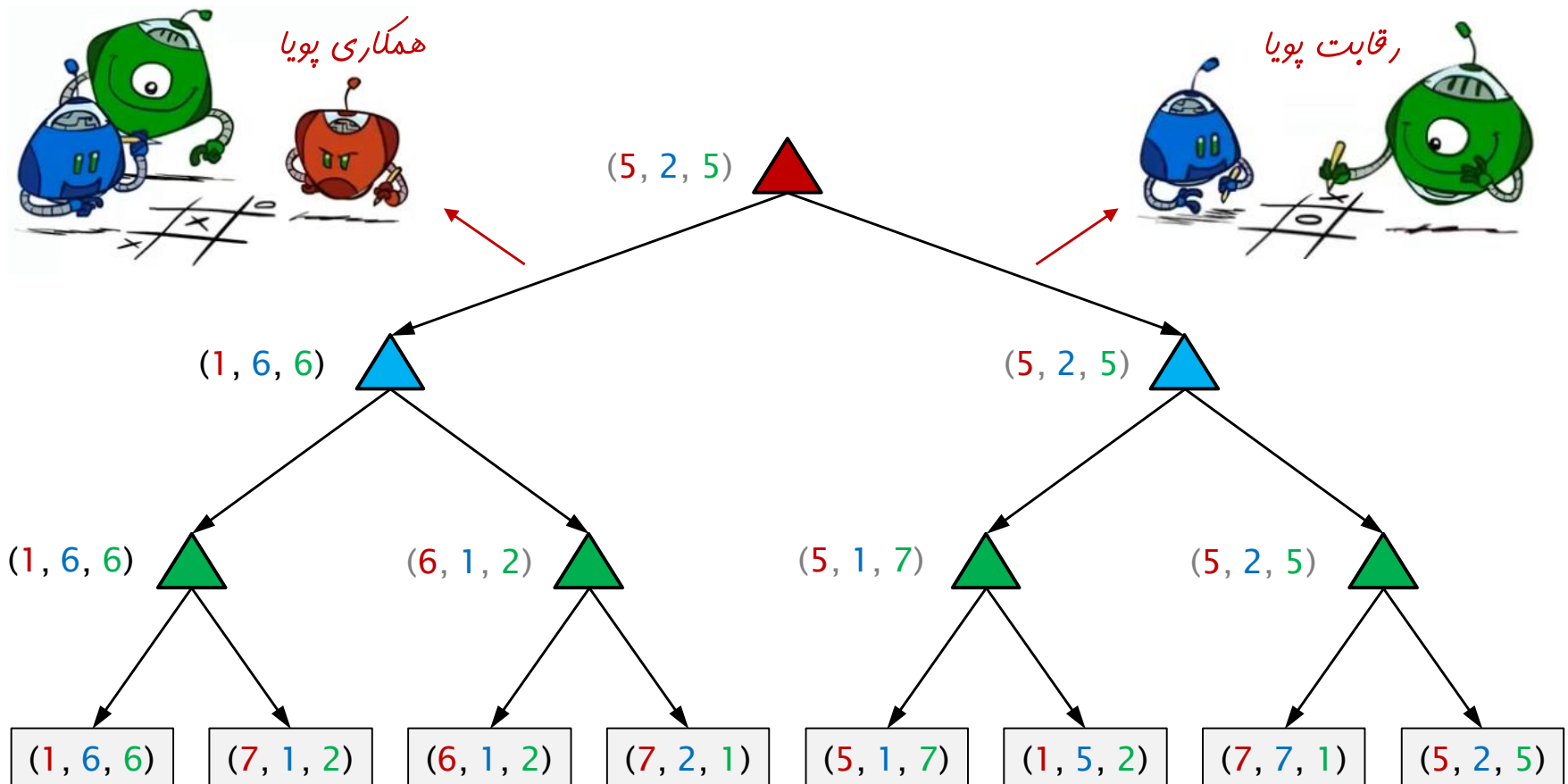
□ اما هرس کردن سخت‌تر می‌شود ...

□ **تاریخچه.** برنامه TDGammon یک جستجو با عمق ۲ انجام می‌دهد، از یک تابع ارزیابی بسیار خوب به همراه یادگیری تقویتی استفاده می‌کند: در سطح قهرمان جهان بازی می‌کند!

بازی‌های چند نفره: تصمیم Minimax

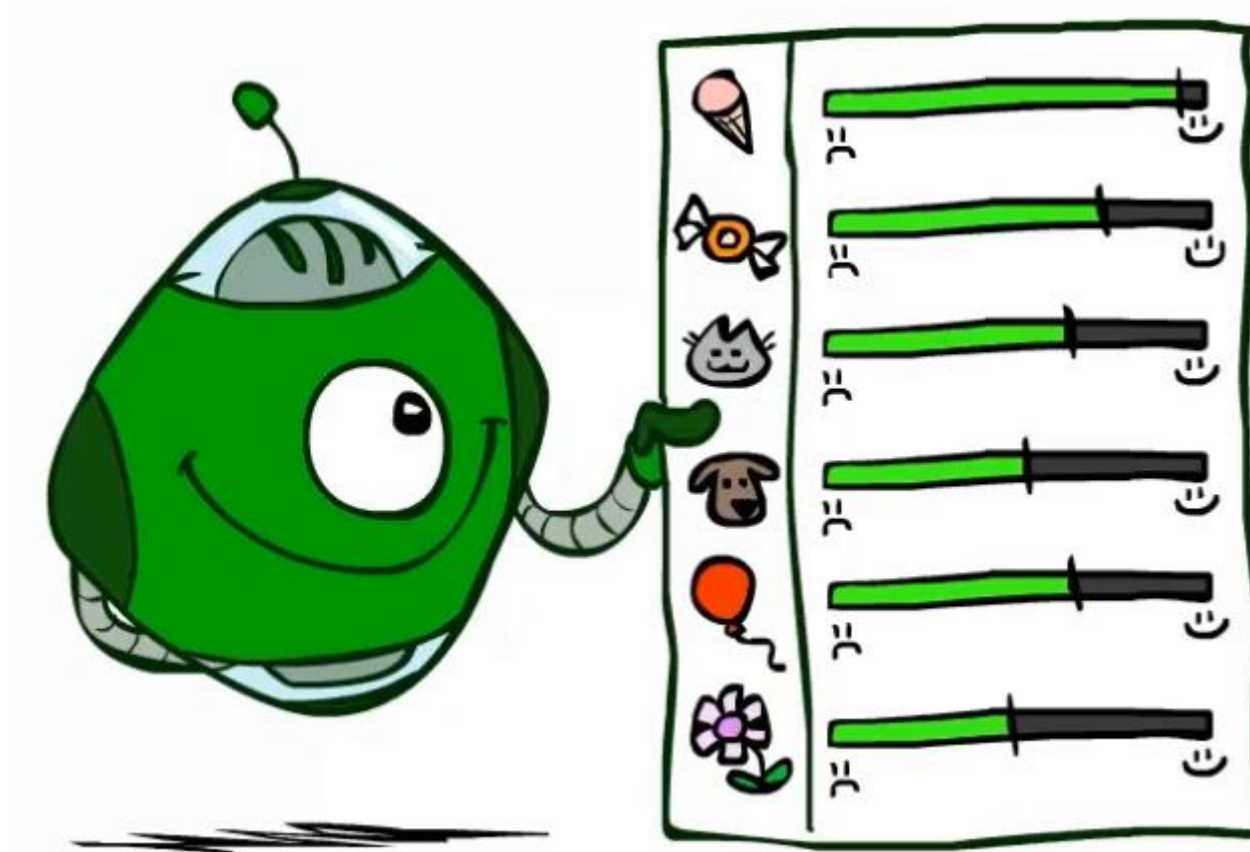
۹۸

□ س. اگر بازی مجموع-صفر نباشد، یا شامل چند بازیکن باشد چه می‌شود؟



سودمندی‌ها

۹۹



بیشینه‌سازی سودمندی مورد انتظار

۱۰۰

□ چرا از مقادیر سودمندی میانگین می‌گیریم؟ چرا همیشه از Minimax استفاده نکنیم؟

□ اصل بیشینه‌سازی سودمندی مورد انتظار.

□ عامل منطقی، با توجه به دانشی که دارد، باید عملی را انتخاب کند که سودمندی مورد انتظار را بیشینه می‌کند.

□ پرسش‌ها.

□ مقادیر سودمندی از کجا می‌آیند؟

□ از کجا می‌دانیم چنین مقادیری وجود دارند؟

□ از کجا می‌دانیم میانگین‌گیری از مقادیر سودمندی معنا دارد؟

□ اگر نتوانیم رفتار (اولویت‌های) را با استفاده از مقادیر سودمندی توصیف کنیم چه؟

سودمندی‌ها

۱۰۱

□ **مقادیر سودمندی.** تابعی از حالات محیط به اعداد حقیقی به گونه‌ای که این اعداد بیانگر میزان مطلوب بودن این حالات برای عامل هستند.



□ **مقادیر سودمندی از کجا می‌آیند؟**

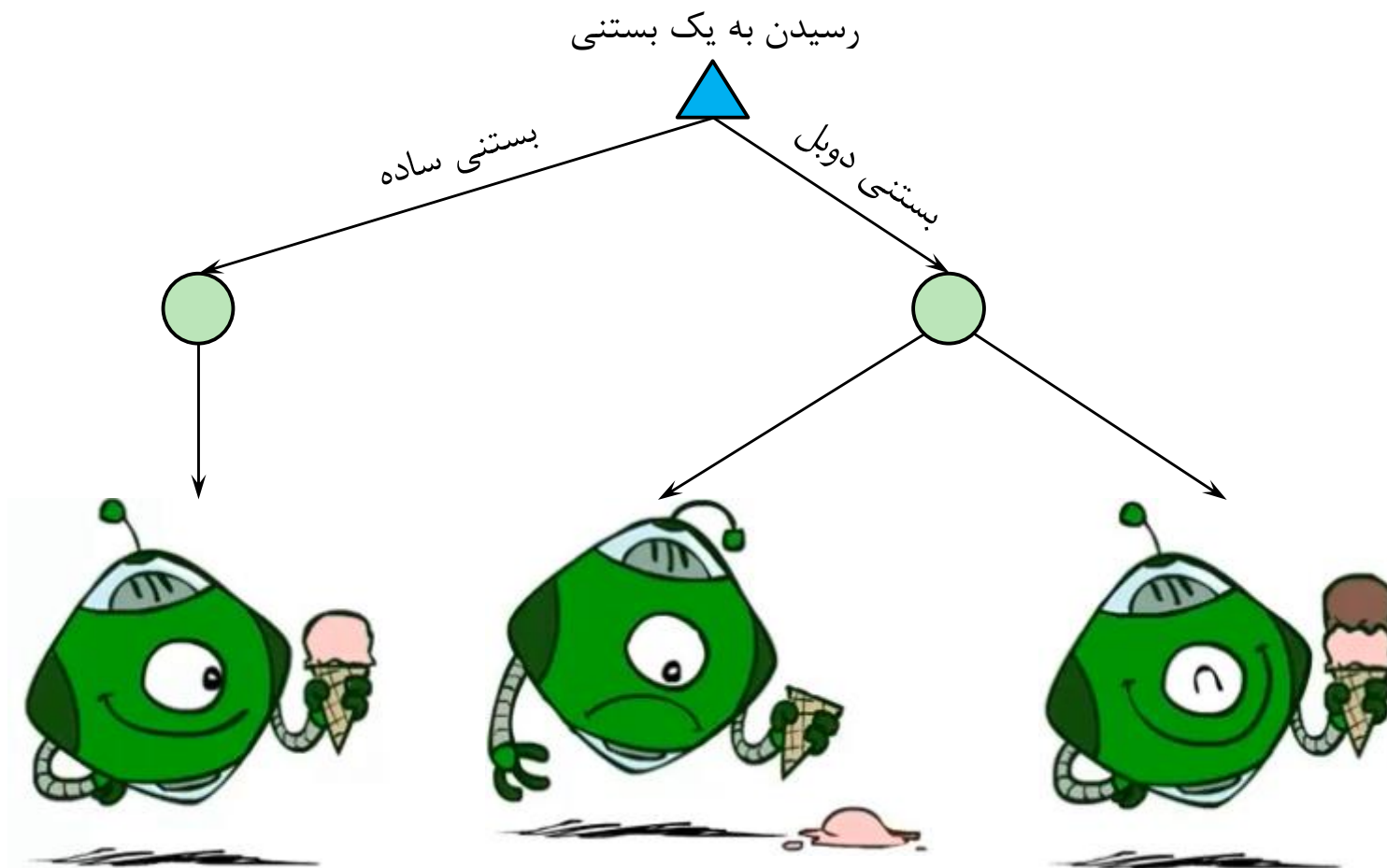
- برای یک بازی، تعریف مقادیر سودمندی می‌تواند ساده باشد (برد: $+1$ و باخت: -1)
- مقادیر سودمندی خلاصه کننده اهداف عامل هستند.
- قضیه: هر گونه اولویت‌های «منطقی» را می‌توان با استفاده از توابع سودمندی خلاصه نمود.

□ **ما فقط سودمندی‌ها را تعریف می‌کنیم و اجازه می‌دهیم رفتار مناسب پدیدار گردد.**

- چرا اجازه نمی‌دهیم عامل‌ها خودشان سودمندی را تعریف کنند؟
- چرا مستقیماً عامل‌ها را بر اساس رفتار برنامه‌نویسی نمی‌کنیم؟

مقادیر سودمندی و عدم قطعیت

۱۰۲



اولویت‌ها

۱۰۳

□ انواع اولویت‌های عامل.

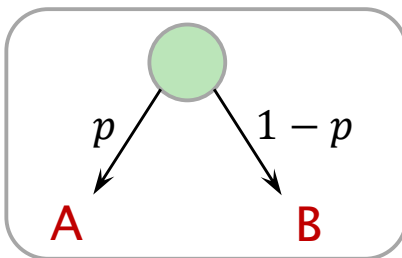
□ اولویت بر روی پیامدها: **A**، **B** و ...

□ اولویت بر روی لاتاری‌ها: موقعیت‌هایی با پیامدهای غیرقطعی

پیامد

A

لاتاری



$$L = [p, \mathbf{A}; (1 - p), \mathbf{B}]$$



اولویت‌ها

۱۰۴

□ انواع اولویت‌های عامل.

□ اولویت بر روی پیامدها: A ، B و ...

□ اولویت بر روی لاتاری‌ها: موقعیت‌هایی با پیامدهای غیرقطعی

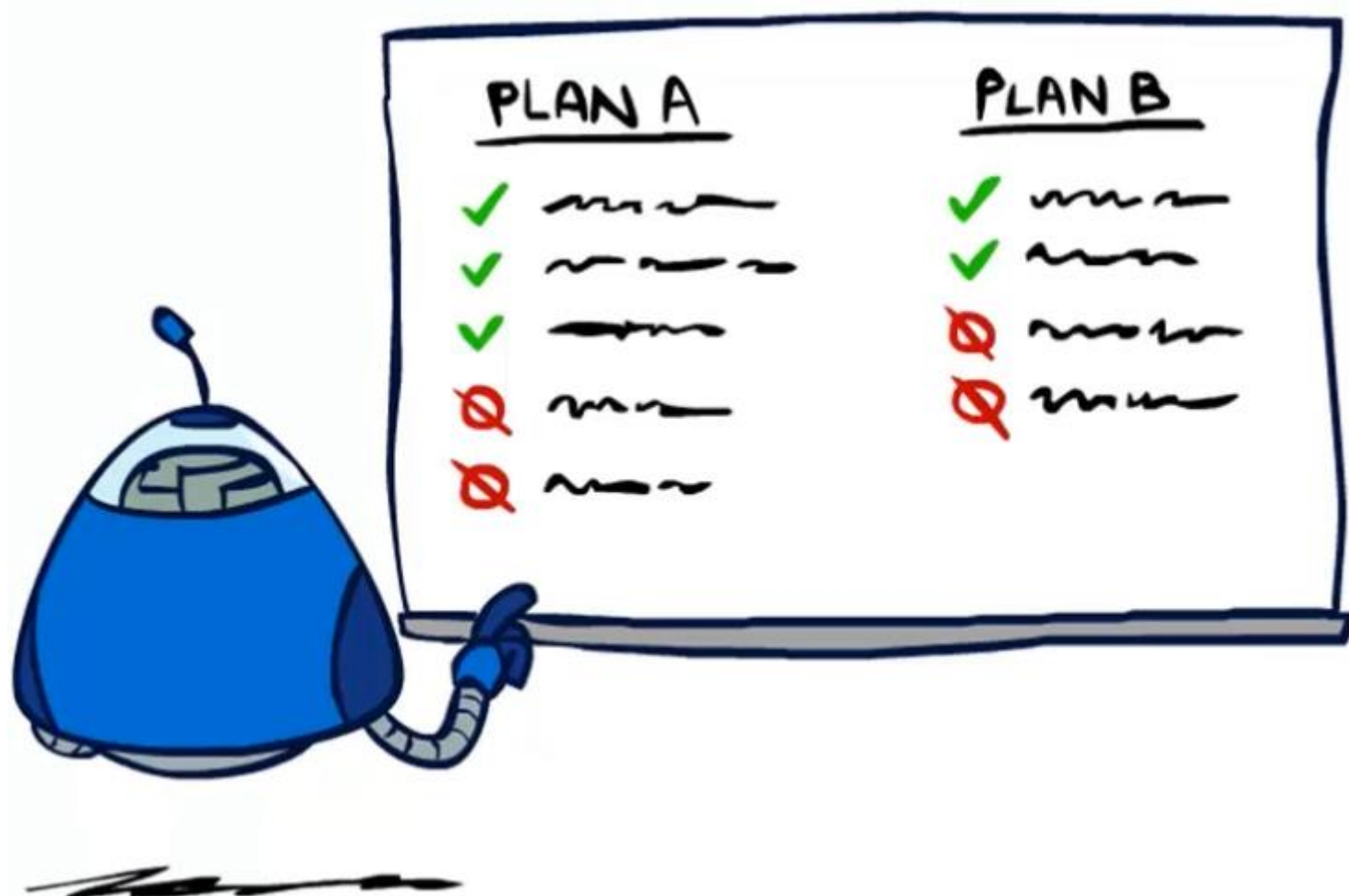
□ نمادها.

□ اولویت: $A \succ B$ عامل پیامد A را به B ترجیح می‌دهد

□ بی تفاوتی: $A \sim B$ برای عامل پیامدهای A و B تفاوتی ندارند

منطقی بودن (عقلانیت)

۱۰۵



اولویت‌های منطقی

۱۰۶

□ اولویت‌های منطقی. اصول موضوعه در منطقی بودن.

$$(A \succ B) \wedge (B \succ C) \Rightarrow (A \succ C)$$

ترانزیتی

$$(A \succ B) \vee (B \succ A) \vee (A \sim B)$$

ترتیب پذیری

$$A \succ B \succ C \Rightarrow \exists p [p, A; 1 - p, C] \sim B$$

پیوستگی

$$A \sim B \Rightarrow [p, A; 1 - p, C] \sim [p, B; 1 - p, C]$$

قابلیت جایگزینی

$$A \succ B \Rightarrow$$

یکنوایی

$$(p \geq q \Leftrightarrow [p, A; 1 - p, B] \succcurlyeq [q, A; 1 - q, B])$$

□ قضیه. [رمزی، ۱۹۳۱؛ وان نیومن و مرگنسترن، ۱۹۴۴]

□ با داشتن هر مجموعه از اولویت‌های منطقی، یک تابع U با مقادیر حقیقی وجود دارد به گونه‌ای که:

$$U(A) \geq U(B) \Leftrightarrow A \succcurlyeq B \quad U([p_1, S_1; p_2, S_2; \dots; p_n, S_n]) = \sum_i p_i U(S_i)$$

□ مقادیر انتساب یافته به وسیله U اولویت‌ها را هم بر روی پیامدها و هم بر روی لاتاری‌ها حفظ می‌کنند.



□ اصل پیشنهادسازی سود مورد انتظار.

□ عملی را انتخاب کن که سودمندی مورد انتظار را بیشینه می‌سازد.

□ توجه. یک عامل بدون استفاده از مقادیر سودمندی و احتمالات هم می‌تواند کاملاً منطقی باشد.

□ مثال. عامل مبتنی بر جدول جستجو برای بازی دوز، عامل واکنشی برای جارو برقی