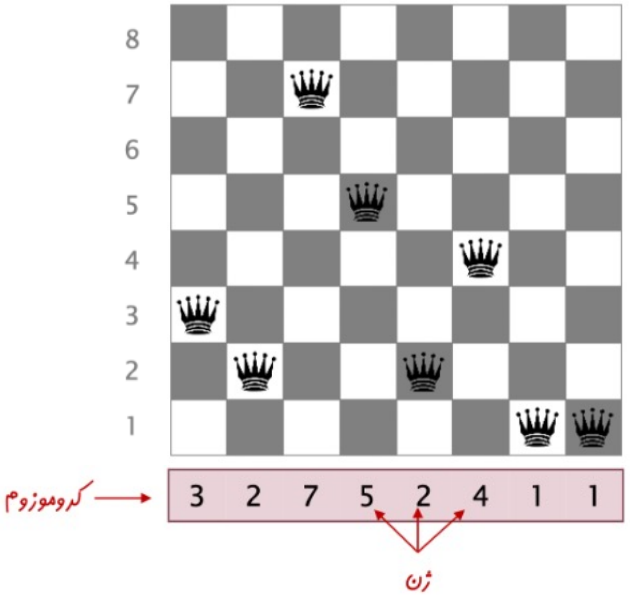
جستجوی محلی
(Local Search)

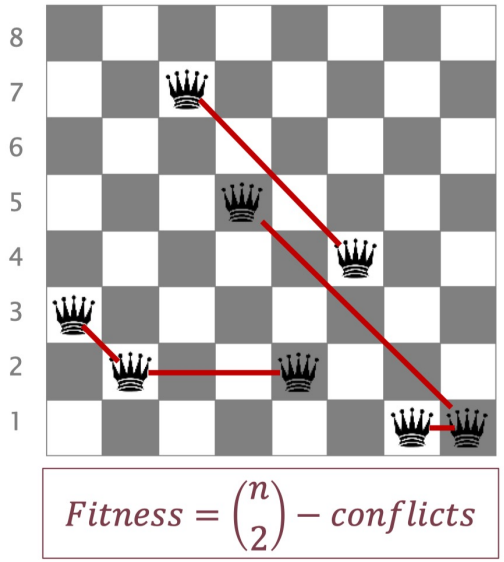
1. الگوریتم تپه نوردی (Hill Climbing)
2. الگوریتم سردسازی شبیه‌سازی شده (Simulated Annealing)
3. الگوریتم گرادیان کاهشی (Gradient Descent)
4. الگوریتم ژنتیک (Genetic Algorithm)

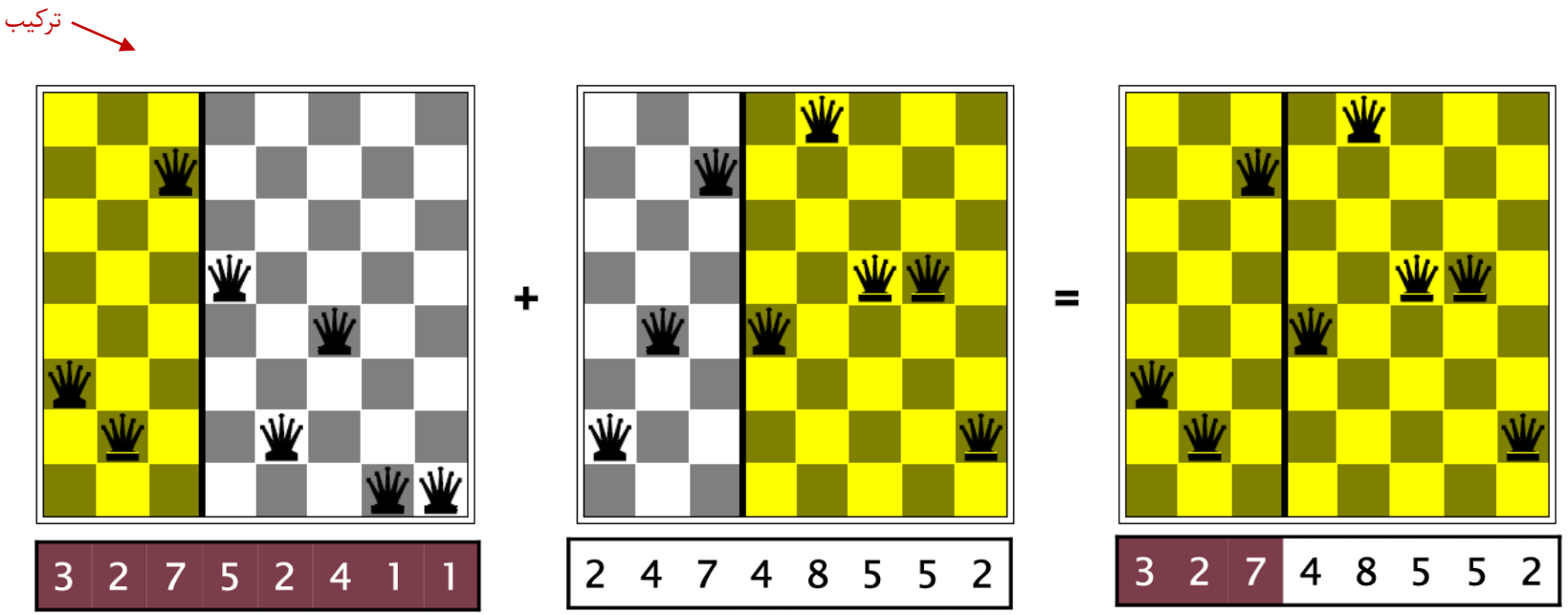
مثال: N- وزیر

طراحی الگوریتم ژنتیک برای N-وزیر:

1. **Encoding:**
   هر کروموزوم یک آرایه‌ی طول N هست که مقدار هر خانه، شماره ردیف وزیر در ستون مربوطه است.
2. **Fitness Function:**
   تعداد تهدیدها بین وزیرها (کمتر = بهتر).
3. **Selection:**
   انتخاب فرضیه‌هایی که تعداد تهدیدهایشان کمتر است.
4. **Crossover:**
   ترکیب دو بخش از والدها برای تولید فرزند (مثلا نصف اول از یکی، نصف دوم از دیگری).
5. **Mutation:**
   تغییر تصادفی مکان یک وزیر در یکی از ستون‌ها.







یادگیری تقویتی
(Reinforcement Learning)

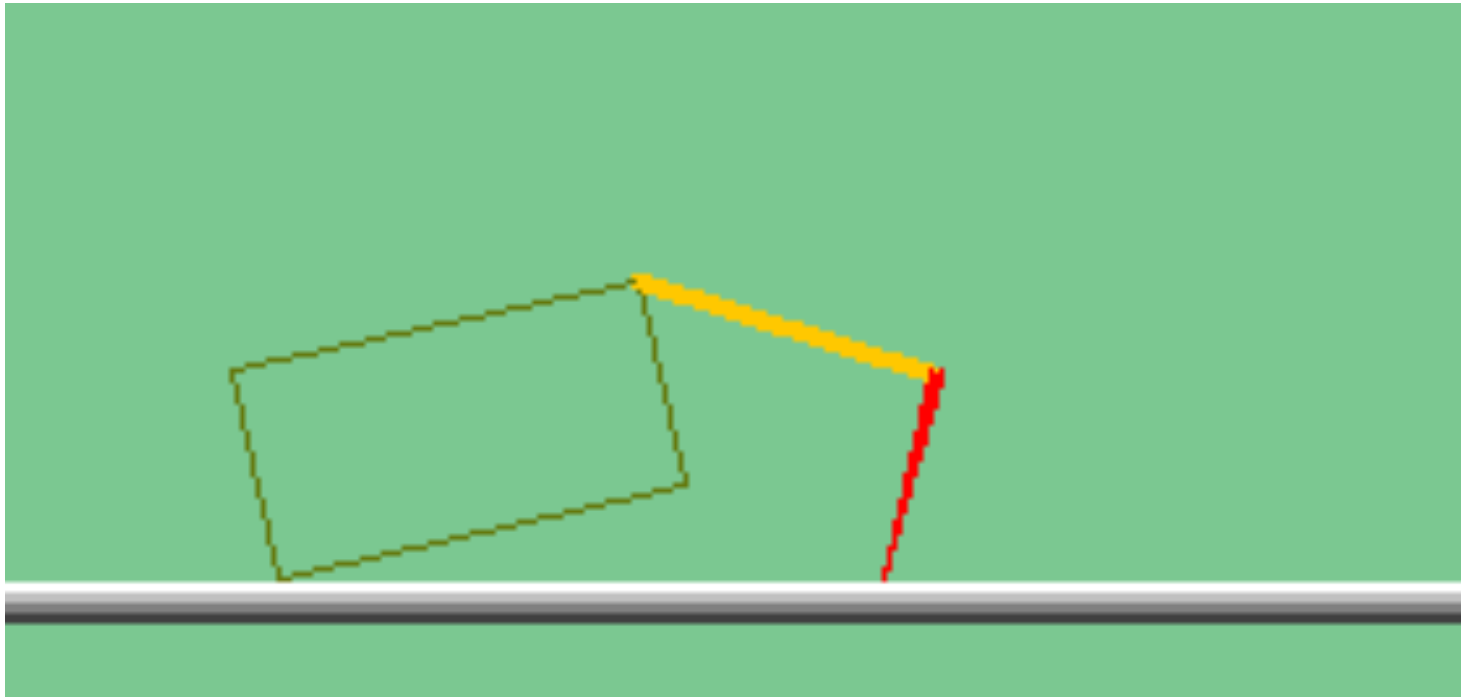
1. فرایند تصمیم مارکوف (Markov Decision Processes)
2. الگوریتم تکرار مقدار (Value Iteration)
3. الگوریتم تکرار سیاست (Policy Iteration)
4. یادگیری تقویتی (Reinforcement Learning)

یادگیری تقویتی (Reinforcement Learning)

- عامل با تعامل با محیط، از طریق آزمون و خطا و دریافت پاداش یا جریمه، یاد می‌گیرد که چگونه دنباله‌ای از اعمال را انجام دهد تا بیشترین پاداش را در بلندمدت بگیرد

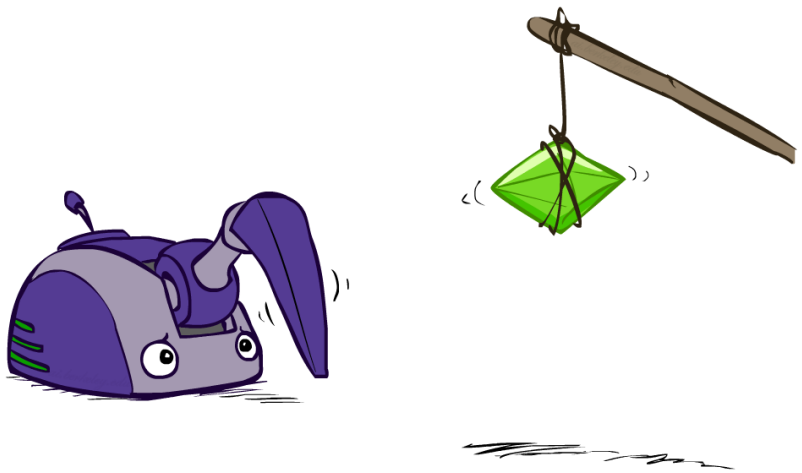
- هدف: انجام کارهایی با برای بیشینه‌سازی پاداش کلی

روبات خزنده!

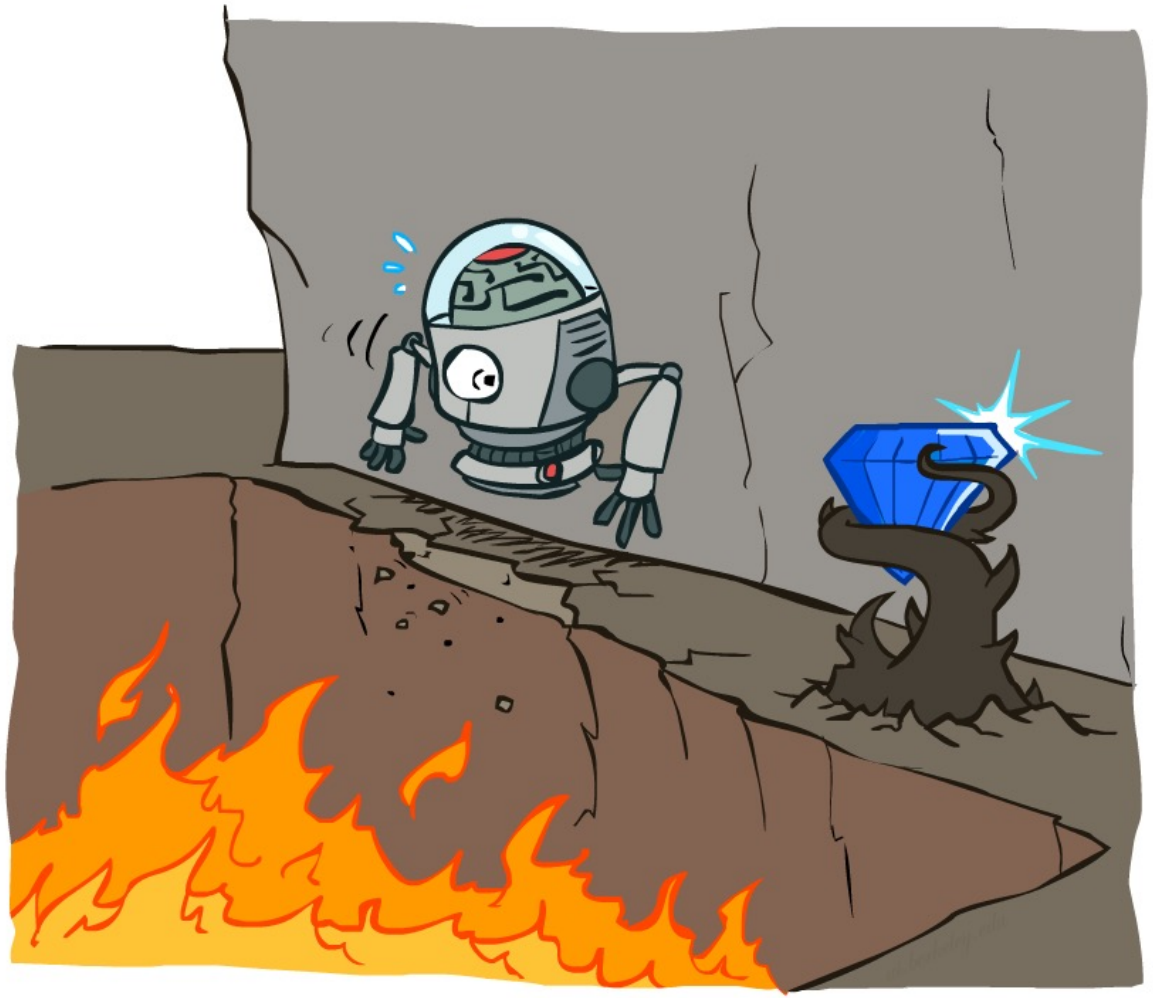


ویدئوی نمایش ربات خزنده

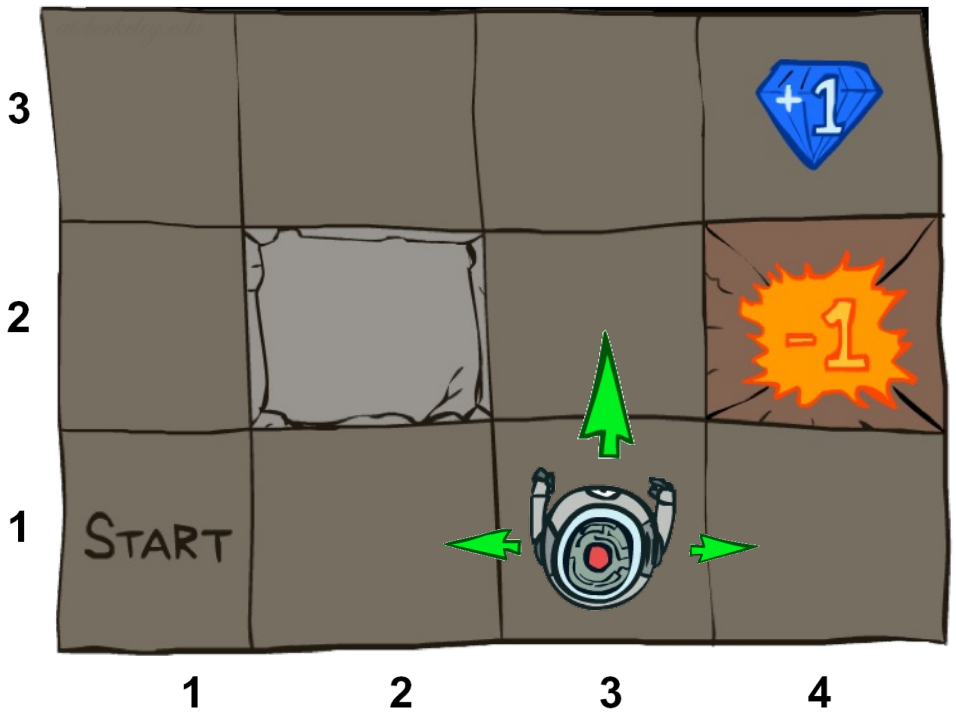
فرایند تصمیم مارکوف
(Markov Decision Processes)

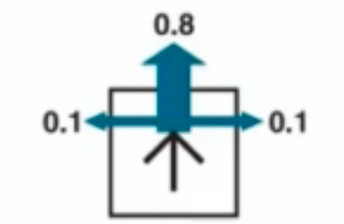


جستجوی تصادفی



مثال: محیط گرید

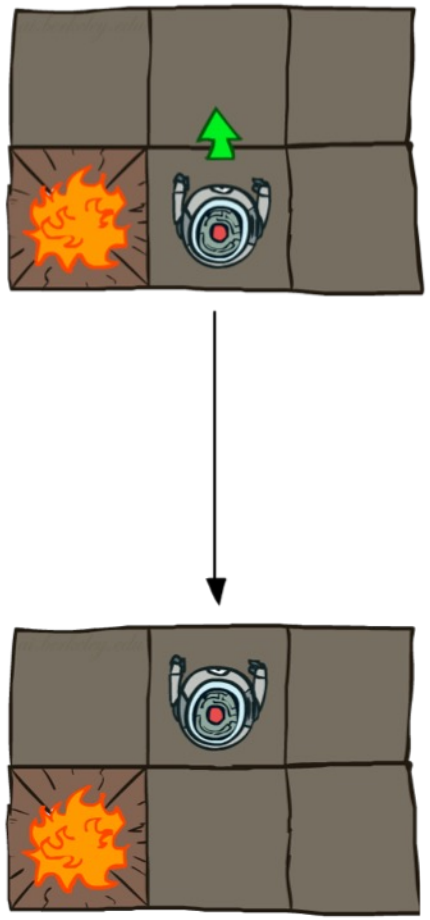




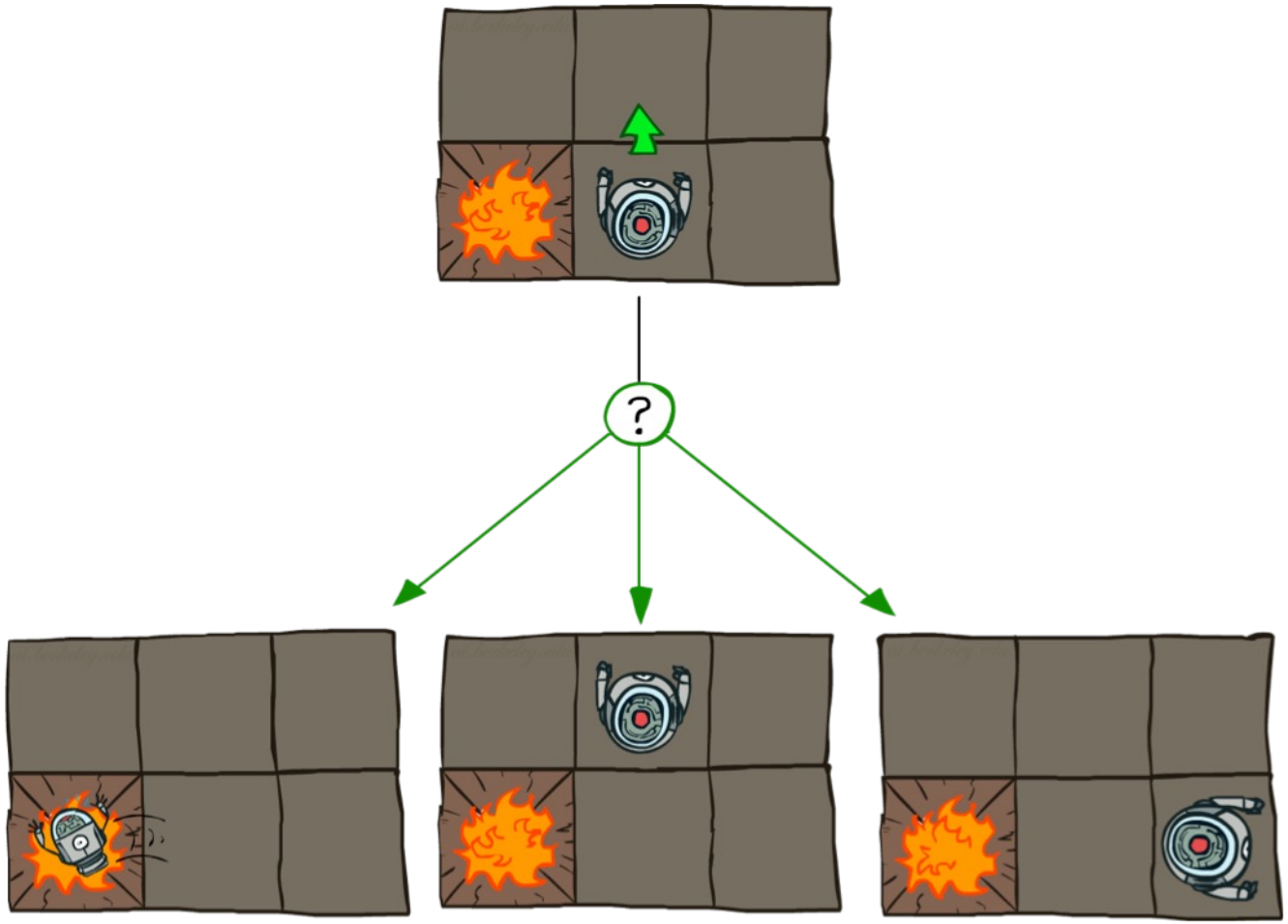
- یک مسئله شبیه به مسیریابی.
  - عامل در یک محیط گرید عمل می‌کند.
  - دیوارها مانع از حرکت عامل می‌شوند.
- حرکت‌های دارای نویز: اعمال همیشه طبق برنامه پیش نمی‌روند
  - ۸۰ درصد مواقع انجام عمل North باعث حرکت عامل به خانه بالایی می‌شود
  - اما ۱۰ درصد مواقع این عمل عامل را به چپ و ۱۰ درصد مواقع عامل را به راست می‌برد.
  - اگر در جهت حرکت عامل دیواری وجود داشته باشد، عامل در مکان فعلی خود باقی میماند.
- عامل پس از انجام هر عمل پاداش دریافت می‌کند
  - پاداش‌های کوچک جهت زنده ماندن پس از هر مرحله (میتواند منفی هم باشد)
  - پاداش‌های بزرگ در انتها (میتواند خوب یا بد باشد)
- هدف: بیشینه کردن مجموع پاداش‌ها

اعمال در محیط گرید

محیط گرید قطعی

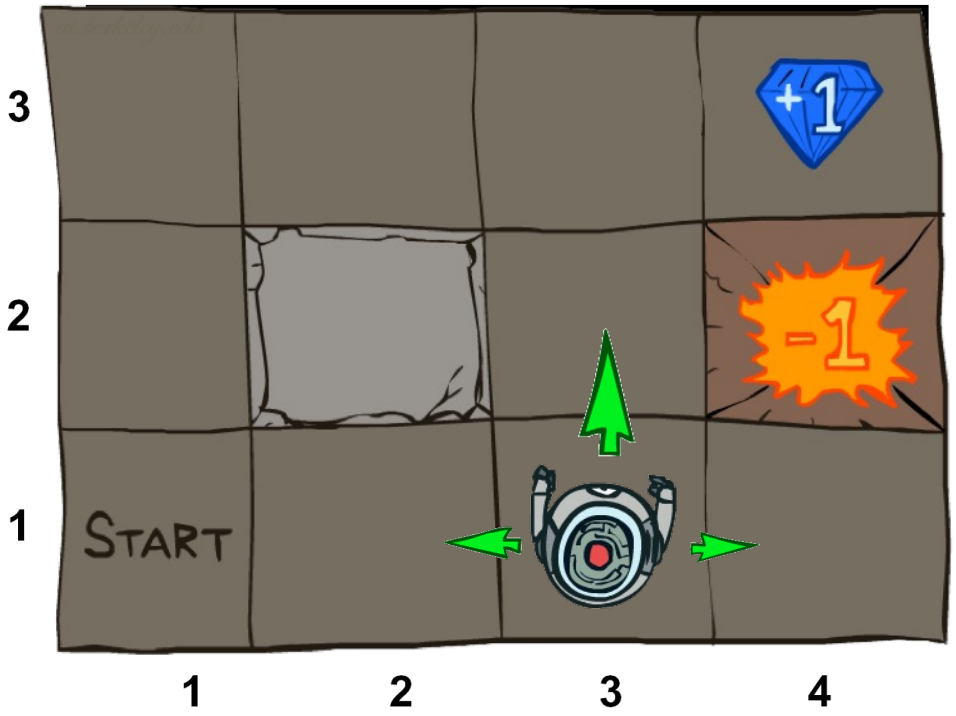


محیط گرید تصادفی



فرایند تصمیم مارکوف (MDP)

➢فرایند تصمیم مارکوف ابزاری است برای بیان(فرموله‌سازی) مسائل از نوع جستجوی غیرقطعی



❑یک MDP به صورت زیر تعریف می‌شود:

- یک مجموعه از حالت‌ها $s \in S$
- یک مجموعه از اعمال $a \in A$
- یک مدل یا تابع تغییر حالت $T(s, a, s')$
  - احتمال رفتن از s به s’ با انجام عمل a . $P(s'| s, a)$
- یک تابع پاداش $R(s, a, s')$ برای هر تغییر حالت
  - گاهی به‌صورت $R(s)$ یا $R(s')$ نمایش داده می‌شود.
- یک حالت شروع و شاید یک حالت پایان

✓ محیط کاملا قابل مشاهده و تصادفی است.

ویدئوی معرفی محیط گرید به‌صورت دستی

ویژگی مارکوف در MDP یعنی چه؟

Andrey Markov
(1856-1922)

- فرایند «مارکوف» معمولا به این معنا است که با داشتن حالت فعلی، حالت بعدی و حالت قبلی از یکدیگر مستقل هستند.
- در فرایندهای تصمیم مارکوف، منظور از «مارکوف» این است که نتیجه‌ی هر عمل تنها به حالت فعلی بستگی دارد.

$$P(S_{t+1} = s' | S_t = s_t, A_t = a_t, S_{t-1} = s_{t-1}, A_{t-1}, \dots S_0 = s_0)$$
$$=$$
$$P(S_{t+1} = s' | S_t = s_t, A_t = a_t)$$

- شبیه به مسئله‌ی جستجو ، که در آن تابع جانشین فقط به وضعیت فعلی بستگی دارد، نه به مسیر طی‌شده تا آن وضعیت (یا همان تاریخچه).

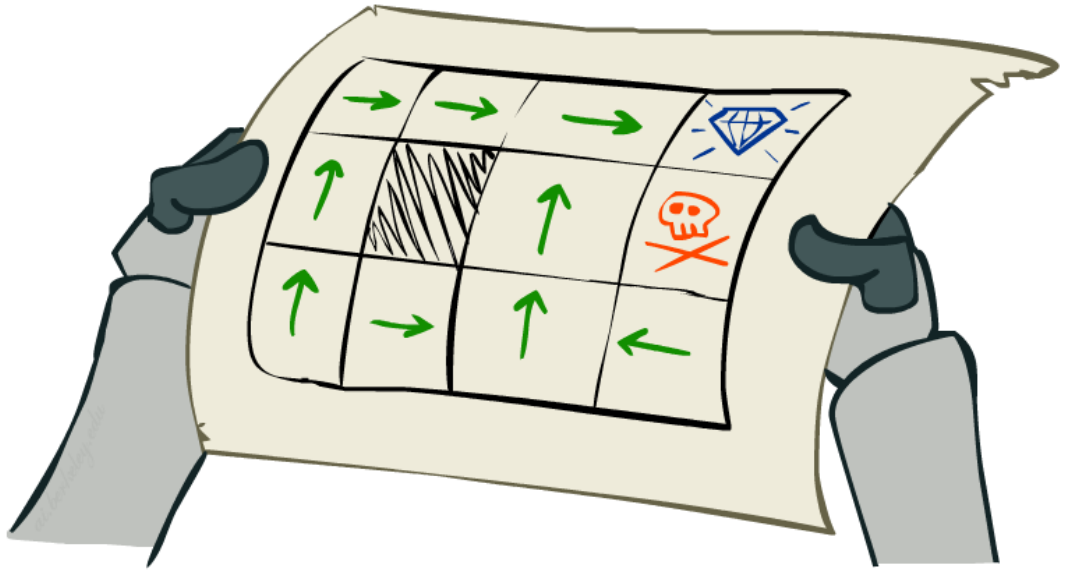
سیاست‌ها (Policies)

❑در مسائل جستجوی تک عاملی قطعی، ما به دنبال یک برنامه بهینه بودیم. (یک دنباله از عملیات از حالت شروع به حالت هدف)

❑در مسائل MDP به دنبال یک سیاست بهینه هستیم:

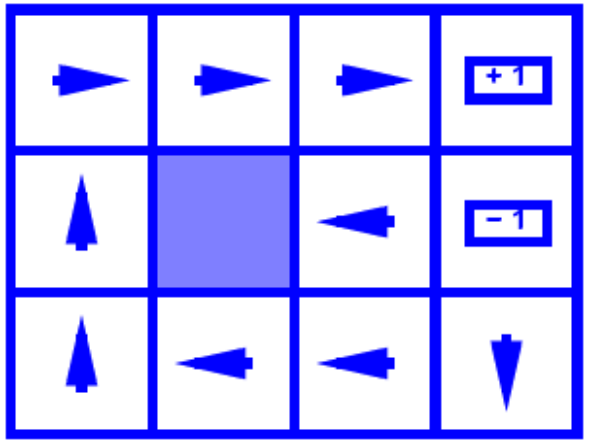
policy $\pi^* : S \rightarrow A$

- سیاست π تعیین کننده‌ی اینست که در هر حالت چه عملی انجام شود.
- سیاستی بهینه است که در صورت دنبال کردن، سودمندی مورد انتظار را بیشینه سازد.
- یک سیاست صریح، بیانگر یک عامل واکنشی (=مقایسه عمل فعلی با لیستی از قوانین ازپیش تعریف شده) است.

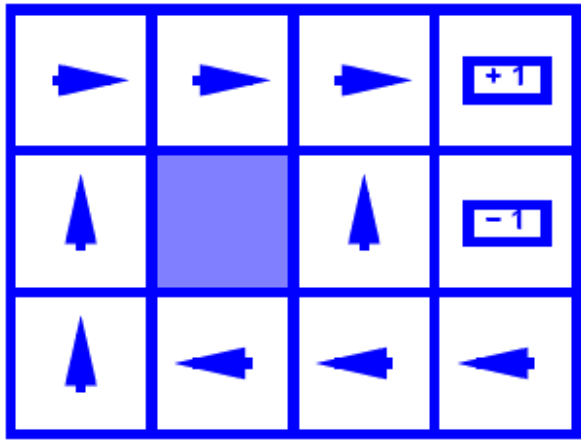


سیاست بهینه وقتی که تابع پاداش برای تمام حالت‌های غیرپایانی (s) $R(s, a, s') = -0.04$ باشد.

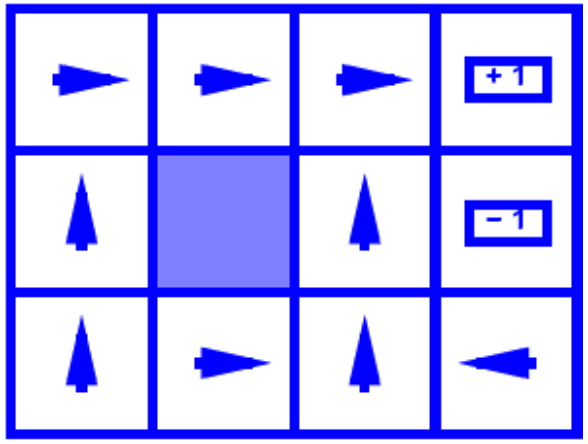
سیاست‌های بهینه



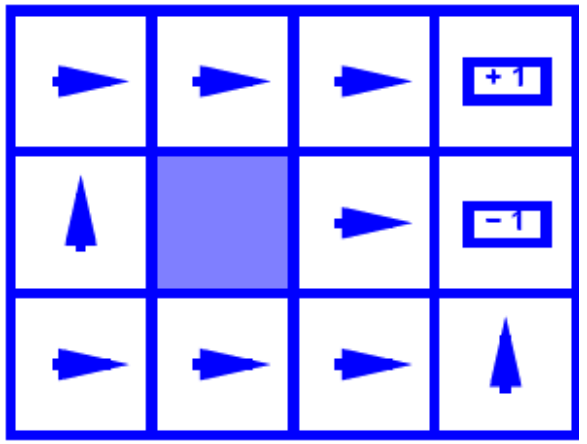
R(s) = -0.01



R(s) = -0.03

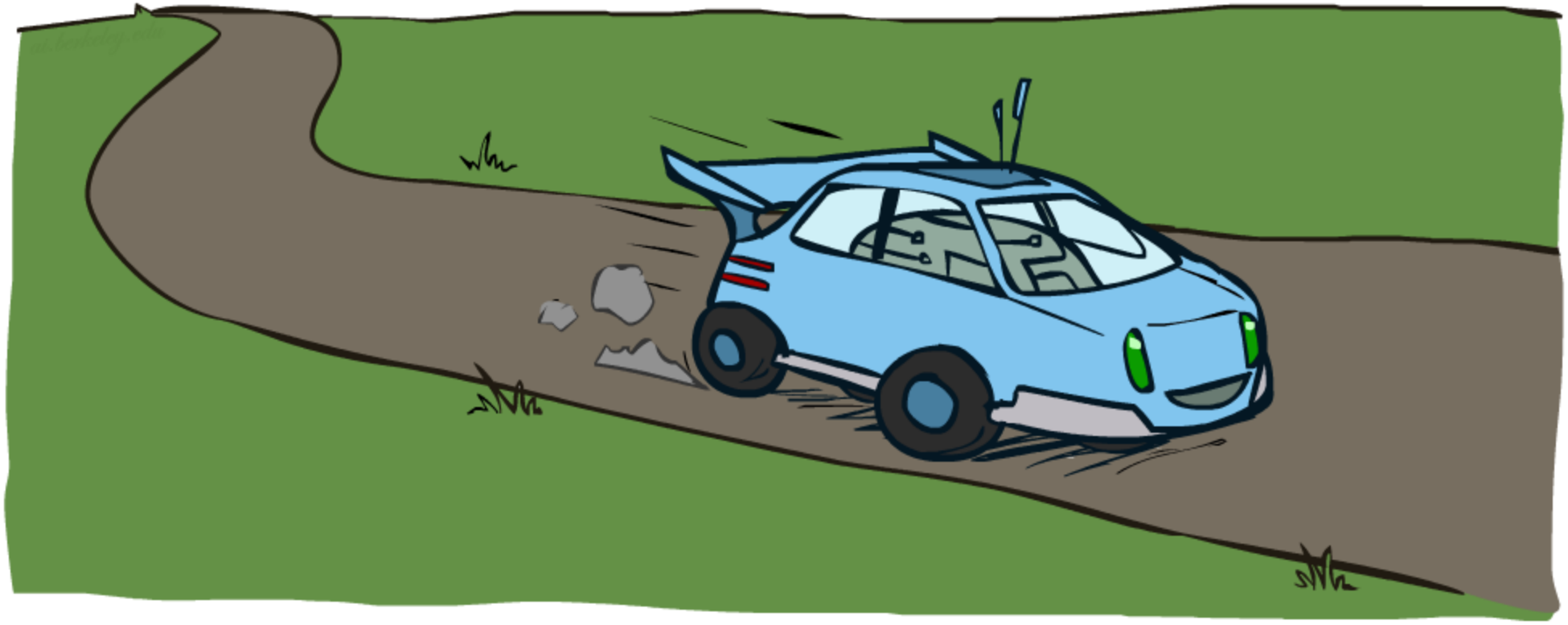


R(s) = -0.4



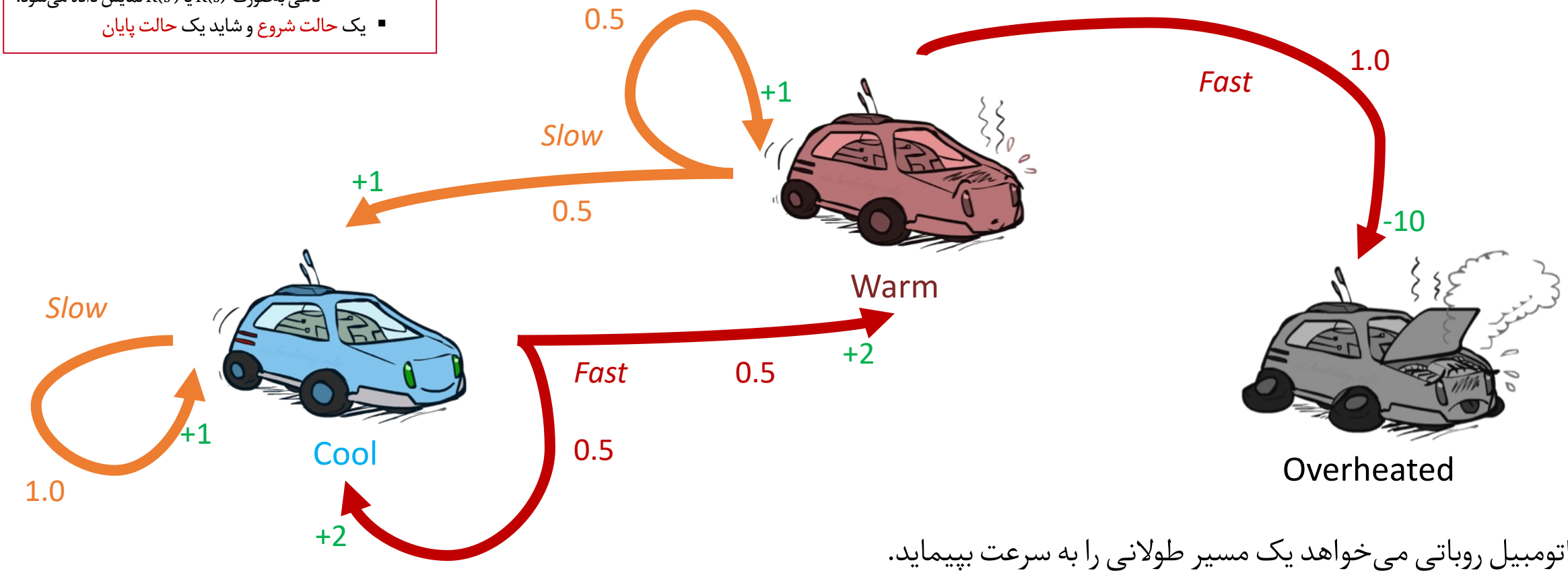
R(s) = -2.0

مثال: مسابقه‌ی اتومبیل رانی



مثال: مسابقه‌ی اتومبیل رانی

❑یک MDP به صورت زیر تعریف می‌شود:
- یک مجموعه از حالت‌ها $s \in S$
- یک مجموعه از اعمال $a \in A$
- یک مدل یا تابع تغییر حالت $T(s, a, s')$
  - احتمال رفتن از s به s' با انجام عمل a . $P(s'|s, a)$
- یک تابع پاداش $R(s, a, s')$ برای هر تغییر حالت
  - گاهی به‌صورت $R(s)$ یا $R(s')$ نمایش داده می‌شود.
- یک حالت شروع و شاید یک حالت پایان

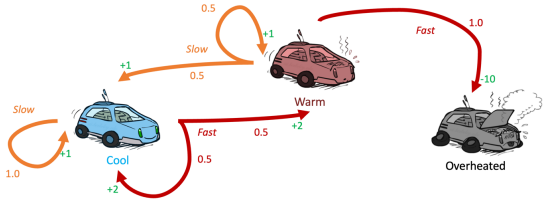


❑یک اتومبیل روباتی می‌خواهد یک مسیر طولانی را به سرعت بپیماید.

❑حالت‌ها: خنک، گرم و جوش

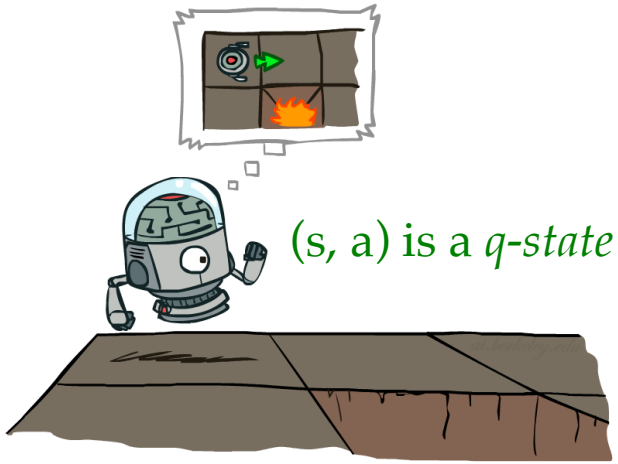
❑اعمال: آهسته، سریع

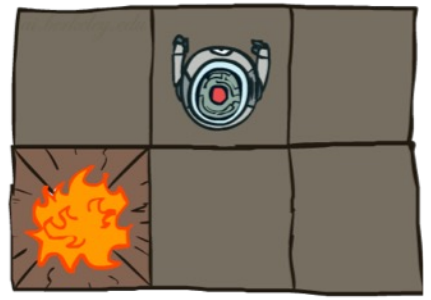
❑سریع رفتن پاداش را دو برابر می‌کند.

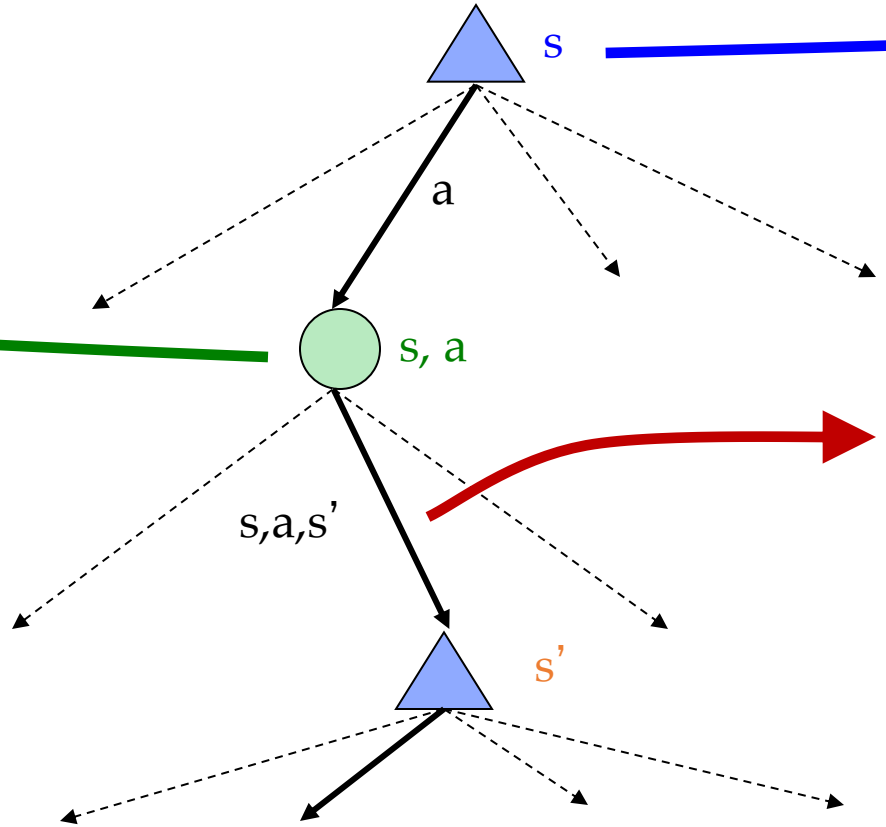


درخت جستجو در مسابقه‌ی اتومبیل رانی

درخت‌های جستجوی MDP در حالت کلی







s is a *state*

(s, a) is a *q-state*

(s,a,s') called a *transition*

T(s,a,s') = P(s' | s,a)

R(s,a,s')