جلسات هفتم، هشتم و نهم

فرآیندهای تصمیم‌گیری مارکوف محدود

یادگیری تقویتی [illegible]

Finite Markov Decision Processing

↳ FMDP

★ اغلب مسائل RL را می‌توان در قالب یک فرآیند تصمیم‌گیری مارکوف (MDP) نشان داد.

★ در مسائل MAB فقط یک حالت وجود داشت و با انجام هر عمل، حالت محیط تغییر نمی‌کرد.

اما در MDP چندین حالت وجود دارد و با انجام عمل محیط از یک حالت به حالت دیگری می‌رود ← بازی Gridworld 5×5

- بازی XO
- مثال RaceTrack
- cliff walking

در این مثال Grid بندی می‌کنیم ← تعداد حالات که برابر تعداد خانه‌های Grid است.

جایزه‌ای که دریافت می‌کنیم هم به Action بستگی دارد و هم State.

★ اگر محیط پیوسته باشد، با گسسته‌سازی و کوانتیزاسیون به FMDP تبدیل می‌شود.

مثال: کنترل آونگ معکوس چرخان

کم و زیادی که به موتور آونگ اعمال می‌کنیم را به‌گونه‌ای کنترل کنیم که آونگ در حال پایدار بماند.

اگر موتور DC 12 ولت داشته باشیم ← سیگنال کنترلی که به موتور اعمال می‌کنیم که آونگ پایدار بماند در بازه‌ی [-12 12] و u(t) قرار دارد.

← بازه کاملاً پیوسته — زاویه آونگ هم در بازه $\theta \in [-180\ 180]$ قرار می‌گیرد که θ و سرعت زاویه‌ای $\dot{\theta}$ هم پیوسته هستند.

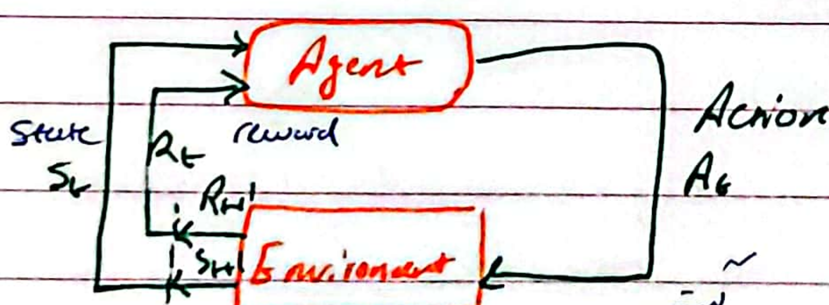
اگر بخواهیم مسئله را با RL حل کنیم باید گسسته‌سازی انجام دهیم. مثلاً:

U: [-12, -11, ..., 1, 11, 12] ⟵ برای θ و $\dot{\theta}$ هم حالات محدودی را در نظر می‌گیریم.

★ این MDP ریاضیات یادگیری تقویتی برای زمانی است که دنیا یک محیط کاملاً ایده‌آل شده باشد.

در چنین حالتی می‌توان پاسخ‌های بهینه را به صورت تحلیلی محاسبه کرد.

t: 0, 1, ...



در این مثال اگر A_t و S_t مشخص باشد، باید بتوانیم S_{t+1} و R_{t+1} را مشخص کنیم.

گسسته ↓

$S_t \in \mathcal{S}$ ← مجموعه‌ی حالات

$A_t \in \mathcal{A}(s)$ — Action‌ها به State هم بستگی دارد.

s.a.m

* در FMDP تمام سیگنال‌ها و حالت‌ها گسسته و دارای چندین مقدار محدود هستند.

$S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, S_3$

دینامیک مدل FMDP:

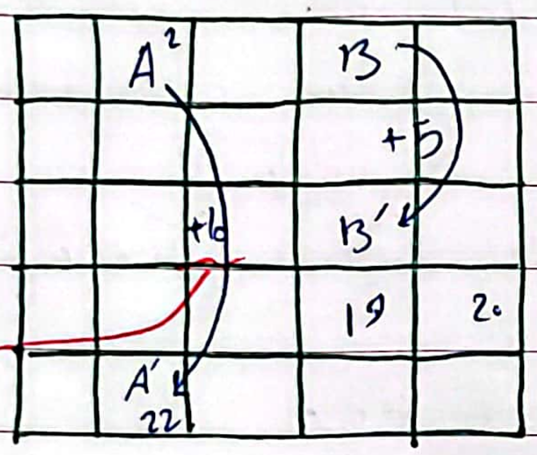
$P: S \times R \times S \times A \rightarrow [0,1]$

ورودی‌ها که در ضرب دکارتی قرار دارند — بازه‌ی خروجی (احتمال)

تابع احتمال $p(s', r | s, a) \doteq Pr\{S_t = s', R_t = r \mid S_{t-1} = s, A_{t-1} = a\} \rightarrow \{0,1\}$

که مشخص می‌کند، اگر در حالت s باشیم و اکشن a را انجام بدهیم، چقدر احتمال وجود دارد که به حالت جدید s' برویم و پاداش r را دریافت کنیم.

چون در RL اغلب محیط‌ها تصادفی هستند ما هم دینامیک را به شکل تابع احتمال تعریف می‌کنیم ولی اگر محیط deterministic بود این احتمال یا ۱ است یا صفر.



$P(22, 9 \mid 2, L) = 0$

اگر در 2 باشیم و اکشن دلخواه انجام بدهیم که به 22 می‌رویم و پاداش 10 خواهد بود نه به صورت 9 و این احتمال صفر می‌شود. ولی اگر شکل زیر باشد:

$P(22, 10 \mid 2, L) = 1$

فرض هم مسائل MDP اینه که برای محیط‌های تصادفی هم می‌تونه استفاده بشه مثل محیطی که R ها هم تصادفی‌اند: $R \in \{0.3/10,\ 0.4/5,\ 0.2/3,\ 0.1/1\}$

اگر محیط تصادفی بود =

$P(22, 10 \mid 2, L) = 0.3$

$P(20, 1 \mid 19, R) = 0$

مثال:

چون در بازی فقط از A به A' و B به B' برویم پاداش می‌گیریم احتمال بالا صفر است چون اگر با اکشن R از 19 حرکت کنیم به 20 می‌رویم.

سیاست بهینه در Gridworld اینه که مدام از A به A' بریم و دوباره برگردیم.

نکته: دینامیک سیستم های MDP یک تابع توزیع احتمال است.

پس از انجام عملی در یک حالت خاص، حالت بعدی و پاداش در داخل عبارت MDP هستند.

$f(x) \rightarrow \int_{-\infty}^{+\infty} f(x)dx = 1$

$$\sum_{s' \in S} \sum_{r \in R} p(s', r | s, a) = 1, \quad for\ all\ s \in S,\ a \in A(s)$$

(s': حالت بعدی، r: پاداش، s: حالت فعلی، a: عمل)

باید عامل طوری طراحی شود که همواره این شرط برقرار باشد.

متغیرهای اصلی ما s' و r است، پس روی s' و r اگر همه احتمالات را جمع کنیم باید یک به دست بیاید.

خاصیت مارکوف: حالت بعدی (s') و پاداش فقط به حالت قبلی (s) و عمل (a) انجام شده در آن وابسته است.

نتیجه: حالت بعدی به تاریخچه وابسته نیست و فقط به حالت قبلی و عمل آن قبلی دارد (آخرین حالت قبلی و اکشن انجام شده)، نه اینکه حالت در ۱۰ تا step قبل چه بوده، برقرار داشته و بر حالت بعدی اثر بگذارد. این فرض صحیح نیست و در واقع حالت های خیلی قبل بر حالت بعدی تاثیری ندارند.

← تابع p کل دینامیک محیط را مشخص می کند.

(1) $p(s', r | s, a)$ ← این را داریم و می‌رسیم به تابع زیر است:

(روی r سیگما sum)

(2) $p(s' | s, a) \doteq Pr\{S_t = s' | S_{t-1} = s, A_{t-1} = a\} = \sum_{r \in R} p(s', r | s, a)$

توابع توزیع حاشیه‌ای ←

یادآوری:
$f(x,y)$
$f_x(x) = \int_{-\infty}^{\infty} f\, dy$

reward ای که به دست می‌آوریم وقتی که در حالت s باشیم و اکشن a را انجام دهیم.

(3) امید ریاضی: $r(s, a) \doteq E[R_t | S_{t-1} = s, A_{t-1} = a] = \sum_{r \in R} r \sum_{s' \in S} p(s', r | s, a)$

اگر محیط determin باشد این expected حذف و با عددی ثابت جایگزین می‌شود.

اگر جزء تصادفی باشد: $R \in [0.5/3, 0.5/6]$ ← امید ریاضی: $0.5 \times 3 + 0.5 \times 6 = 4.5$ اگر هزار بار این Action را انجام دهیم 4.5 می‌گیریم.

یادآوری:
$E[x] = \sum x f_x(x)$

④ $r(s,a,s') = E[R_t \mid S_{t-1}=s, A_{t-1}=a, S_t=s'] = \sum_{r \in R} r \frac{p(s',r \mid s,a)}{p(s' \mid s,a)}$ } قانون بیز ★

$p(r \mid s,a,s')$ →

با اینها از چهار تابع برتر ① می‌توانیم مسائل RL را حل کنیم.

ربات جمع‌آوری زباله:

حالت‌ها (براساس وضعیت باتری): {low, high}

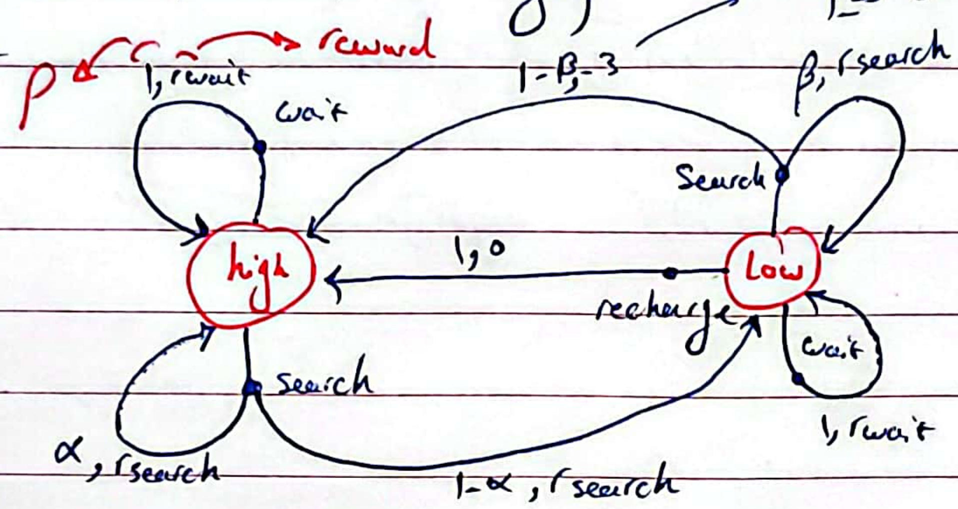
عمل‌ها: {search, wait, recharge}

اگر باتری شارژش بلند باشد چرا شارژش می‌کنیم.

$a \in A(s)$

$A(high) = \{s, w\}$

$A(low) = \{s, w, re\}$



تعیین برتر:

$P(high, r_s \mid high, s) = \alpha$

$P(low, r_s \mid high, s) = 1-\alpha$

جدول:

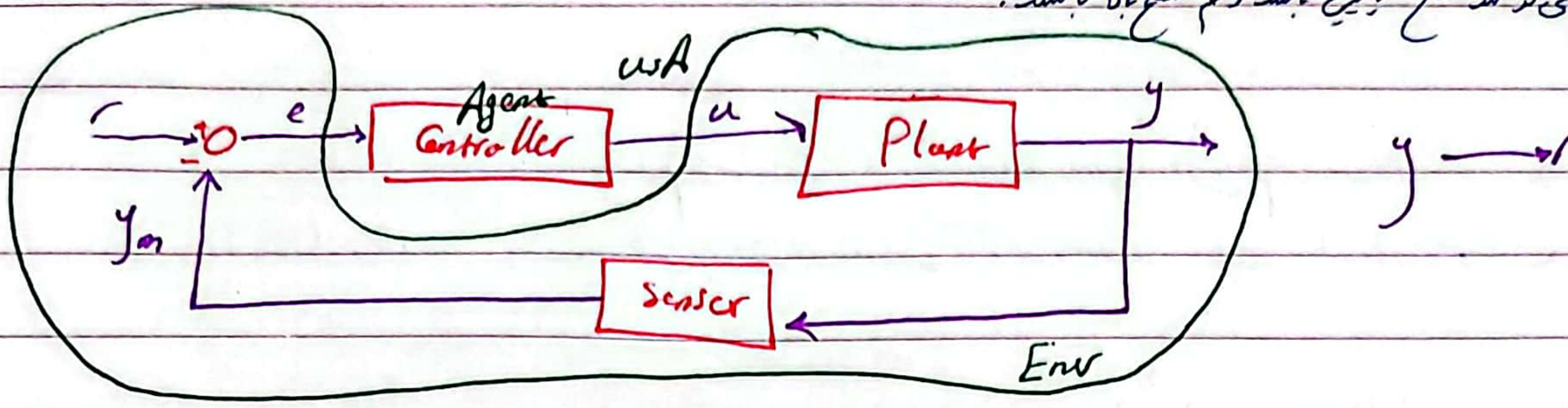
s	a	s'	$p(s' \mid s,a)$	$r(s,a,s')$
high	search	high	α	rsearch
"	"	low	$1-\alpha$	"
low	"	high	$1-\beta$	-3
"	"	low	β	rsearch
high	wait	high	1	rwait
"	"	low	0	–
low	"	high	0	–
"	"	low	1	rwait
"	recharge	high	1	0
"	"	low	0	–

ادامه فرآیندهای تصمیم مارکوف محدود

★ در MDP لازم نیست متغیر t بیانگر زمان واقعی باشد و می‌تواند مراحل متوالی تصمیم‌گیری و انجام عمل را نشان دهد.
← time step

t: 0 →(1s) 1 →(10s) 2

عمل‌ها در MDP هم می‌توانند سطح پایین باشند و هم سطح بالا باشند.



یک حالت دیگر هست که می‌توانیم بیاییم کل سیستم حلقه بسته را به عنوان Env در نظر بگیریم. در اینجا A را (setpoint) در نظر می‌گیریم. حالا یک RL بیاید این را تعیین می‌کند.

★ در یادگیری تقویتی **اهداف** عامل در پاداش‌ها نوشته می‌شود. هدف عامل انتخاب عمل‌هایی است که **بیشینه پاداش در درازمدت** بدست (ماکزیمم) شود. انتخاب پاداش‌ها باید به دقت انجام شود تا منجر به نتایج غیرهدفمند نشوند ← ارتباط با عملیات نوشته شده حالت پاداش است.
نتایج دانش‌های قبلی از طریق پاداش‌ها به عامل داده می‌شوند ← پاداش‌ها برای بیان هدف هستند نه راه رسیدن به هدف.

Return → Cumulative Reward

به پاداش تجمعی بازگشت می‌گوییم و هدف عامل ماکزیمم کردن آن است.

Return G_t = جمع پاداش‌های دریافتی بعد از t

$S_0, A_0, R_1, S_1, A_1, \dots$

$G_t = R_{t+1} + \dots + R_T$

$G_0 = R_1 + R_2 + \dots$

★ عملیات دوره‌ای (Episodic Task): مثلاً بازی شطرنج و ... ← دارای حالت **نهایی** هستند. (Terminal State)

★ عملیات پیوسته (Continuous Task): مثلاً کنترل فرآیندهای صنعتی $T = \infty$
Terminal State ندارند.

در عملیات دوره‌ای معمولاً برای هر دوره حالت اولیه به صورت کاملاً تصادفی انتخاب می‌شود.

برای عملیات پیوسته اغلب بازگشت **بی‌نهایت** می‌شود ← با تعریف نرخ تخفیف **Discount Rate** (Discount Factor) به پاداش‌های اخیر اهمیت بیشتری داده می‌شود.

s.a.m

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} = \sum_{k=t+1}^{\infty} \gamma^{k-t-1} R_k$$

عامل سعی می‌کند سریعتر به اهداف برسد و سعی می‌کند پاداش‌های طولانی مدت را تحویل نگیرد.

ضریب تخفیف:

$\gamma = 0$ → عدم توجه به آینده و نقش پاداش فعلی پررنگ می‌شود.

$\gamma = 1$ → (دوراندیشی) تمامی پاداش‌های دریافتی در آینده اهمیت یکسان دارند → حساب می‌شود.

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots$$
$$= R_{t+1} + \gamma (R_{t+2} + \gamma R_{t+3} + \dots)$$
$$G_t = R_{t+1} + \gamma G_{t+1}$$

$$R_i = 1 \longrightarrow G_t = \sum_{k=0}^{\infty} \gamma^k = \frac{1}{1-\gamma}$$

اگر از k_0 شروع شود: $\dfrac{\gamma^{k_0}}{1-\gamma}$

برای حل این مسئله باید شروع کنیم برسیم به G_1 مثلا.

سیاست و تابع ارزش:

اغلب الگوریتم‌های یادگیری تقویتی **توابع ارزش** را تخمین می‌زنند.

تابع ارزش حالت $V(s)$: نشان دهنده میزان ارزشمندی حالت s

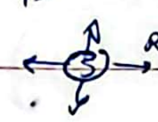
تابع ارزش حالت-عمل $Q(s,a)$: نشان دهنده میزان ارزشمندی انجام عمل a در حالت s

ارزش حالت s نسبت به سیاست مشخص:
$$v_\pi(s) \doteq E_\pi[G_t \mid S_t = s] = E_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t = s\right], \text{ for all } s \in S$$

$$q_\pi(s,a) \doteq E_\pi[G_t \mid S_t = s, A_t = a] = E_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t = s, A_t = a\right]$$

که بین اکشن‌های a بر اساس سیاست π توزیع شده‌اند.

روابط بازگشتی در اغلب الگوریتم‌های یادگیری تقویتی وجود دارند. (شروع سازه‌ی)



$\pi(R|s) = \frac{1}{4}$

$$v_\pi(s) = E_\pi[G_t \mid S_t = s]$$

$$= E_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s]$$

$$= \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r \mid s, a)\,[r + \gamma E_\pi[G_{t+1} \mid S_{t+1} = s']]$$

که این حالت شده که احتمال اتفاق افتادنش s' چیه؟ همین عبارت که زیرش خط کشیدیم ضرب شده

در اینجا یعنی با در نظر گرفتن a های مختلف حاصل جمع رو در نظر بگیریم sum که بیرونی هم گذاشته شود.

$$= \sum_a \pi(a|s) \sum_{s',r} p(s', r \mid s, a)\,[r + \gamma v_\pi(s')] \quad \text{for all } s \in S$$

در حالت‌های احتمالی انتخاب a چقدر است؟

* معادله بلمن برای v_π

* معادلات بلمن یک دسته n معادله n مجهول خطی است.

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r \mid s, a)\,[r + \gamma v_\pi(s')] \quad \text{for all } s \in S$$

$$E[x+y] = E[x] + E[y]$$

یادآوری: امید ریاضی => $\sum_x x\,p(x)$ ، **اگر ۲ متغیره بود:** $\sum_{x,y} x\,p(x,y)$

در یادگیری تقویتی دنبال سیاستی هستیم که **بیشترین پاداش را در طولانی مدت** تولید کند.

برای مسائل **MDP** با داشتن دینامیک آن می‌توان سیاست بهینه را به شکل تحلیلی محاسبه نمود.

سیاست π بهتر از سیاست π' است اگر برای تمام حالت‌ها منجر به **ارزش حالت بیشتر** شود.

$$\pi \geq \pi' \quad \text{if and only if} \quad v_\pi(s) \geq v_{\pi'}(s) \quad \forall s$$

سیاست بهینه تنها وجود دارد ولی لزوما یکتا نیست.

$$v_*(s) \doteq \max_\pi v_\pi(s) \quad \text{for all } s \in S$$

← تمام سیاست‌های ممکن

اون π که بیشترین ارزش را به ازای حالت ایجاد می‌کند انتخاب می‌کنیم.

* تابع ارزش حالت عمل بهینه

$$q_*(s,a) \doteq \max_\pi q_\pi(s,a) \quad \text{for all } s \in S \text{ and } a \in A(s)$$

$$S_t \xrightarrow[R_{t+1}]{a} S_{t+1}$$

ارتباط بین توابع ارزش:

$$q_*(s,a) = E[R_{t+1} + \gamma v_*(S_{t+1}) \mid S_t = s, A_t = a]$$

$$G_t = R_{t+1} + \gamma G_{t+1}$$

یعنی از t تا t+1 دلخواه ولی از آنجا به بعد بهینه انجام بشه ولی برای بعد از t+1 هم بهینه هستند.

سیاست بهینه π^*:

$$\pi^*(s) = \arg\max_a q_*(s,a)$$

$$= \arg\max_a \sum_r \sum_{s'} p(s', r \mid s, a)\,[r + \gamma v_*(s')]$$

← دینامیک محیط

اگر بخواهیم از v_* استفاده کنیم باید دینامیک معلوم باشد.

بهینه

$v_*(s) = \max_{a \in A(s)} q_{\pi_*}(s,a)$

$= \max_a E_{\pi_*}[G_t \mid S_t = s, A_t = a]$

$= \max_a E_{\pi_*}[R_{t+1} + \gamma G_{t+1} \mid S_t = s, A_t = a]$

$= \max_a E[R_{t+1} + \gamma v_*(S_{t+1}) \mid S_t = s, A_t = a]$

$= \max_a \sum_{s',r} p(s',r \mid s,a)[r + \gamma v_*(s')]$

معادلات بهینگی بلمن

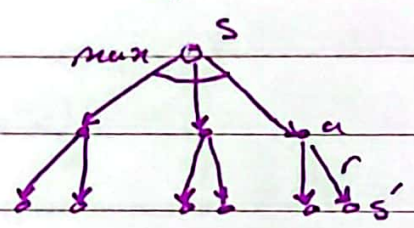
$E_{\pi_*}[\gamma G_{t+1}]$ هم میتوانی بنویسی

$q_*(s,a) = E[R_{t+1} + \gamma \max_{a'} q_*(S_{t+1}, a') \mid S_t = s, A_t = a]$

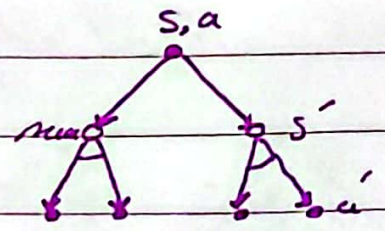
$= \sum_{s',r} p(s',r \mid s,a)[r + \gamma \max_{a'} q_*(s',a')]$

چیست که به دارم:

backup diagram: v_*



q_*



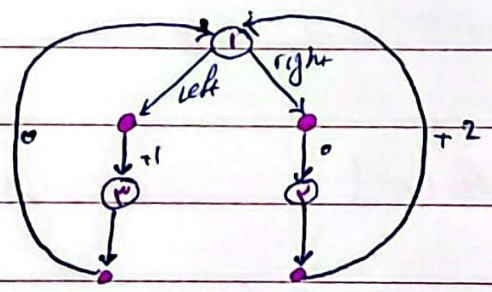
اکشن گره‌ای تصمیم

استیت گره‌ای تصادفی

$\pi_*(s) = \arg\max_a q_*(s,a)$

مثال: برای نرخ تخفیف های ۰، ۰.۵، و ۰.۹ سیاست بهینه را تعیین کنید.

چون ساده است از همان رابطه return می‌نویسیم بلمن نیازیست.



$G_0 = R_1 + \gamma R_2 + \gamma^2 R_3 + \dots$ ، Right ای

$= 0 + \gamma \times 2 + \gamma^2 \times 0 + \dots$

$= 2\gamma + 2\gamma^3$

$v_\pi(1)$ $G_0 = 2\gamma(1 + \gamma^2 + \dots) = \frac{2\gamma}{1-\gamma^2}$

$G_0 = 1 + \gamma \times 0 + \gamma^2 \times 1 + \dots$ left ای

$= 1 + \gamma^2 + \gamma^4 = \frac{1}{1-\gamma^2}$

γ	Right, $\frac{2\gamma}{1-\gamma^2}$	left, $\frac{1}{1-\gamma^2}$
0	0	1
0,5	4/3	4/3
0,9	$\frac{1,8}{1-0,9^2}$	$\frac{1}{1-0,9^2}$