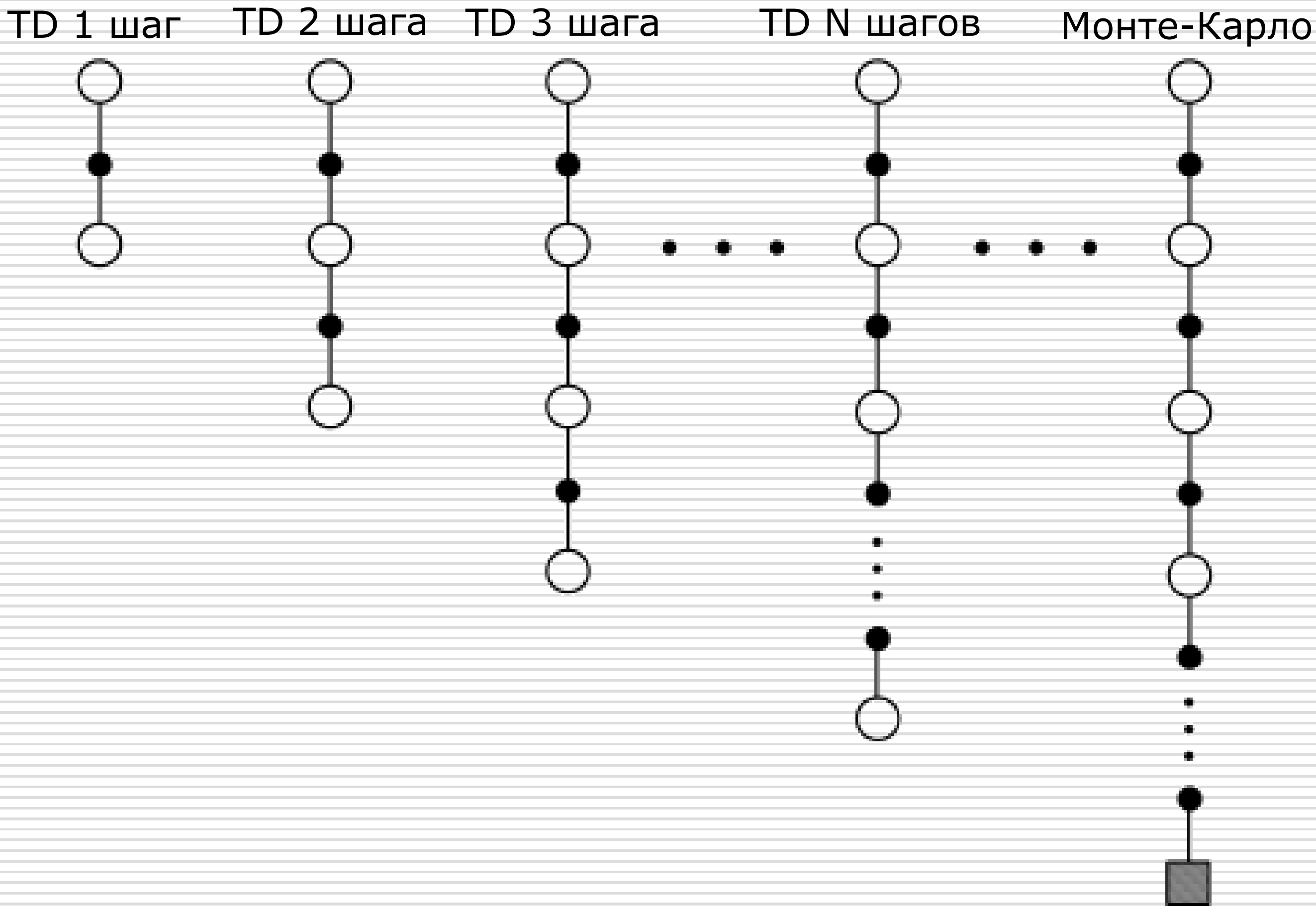
Методы решения задач обучения с подкреплением

Следы преемственности

Методы временных разностей и методы Монте-Карло



Возвраты методов временных разностей и Монте-Карло

- Монте-Карло, полный возврат
  $$R_t^{(n)} = r_{t+1} + \underline{\gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots + \gamma^{T-t-1} r_T}.$$
- TD(0), одношаговый возврат
  $$R_t^{(n)} = r_{t+1} + \underline{\gamma V_t(s_{t+1})}.$$
- Двухшаговый возврат
  $$R_t^{(n)} = r_{t+1} + \underline{\gamma r_{t+2} + \gamma^2 V_t(s_{t+2})}.$$
- N-шаговый возврат
  $$R_t^{(n)} = r_{t+1} + \underline{\gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots + \gamma^{n-1} r_{t+n} + \gamma^n V_t(s_{t+n})}.$$

Многошаговые методы временных разностей

- N-шаговое обновление

$$\Delta V_t(s_t) = \alpha\left[R_t^{(n)} - V_t(s_t)\right],$$

- Онлайн обновление

$$V_{t+1}(s) = V_t(s) + \Delta V_t(s)$$

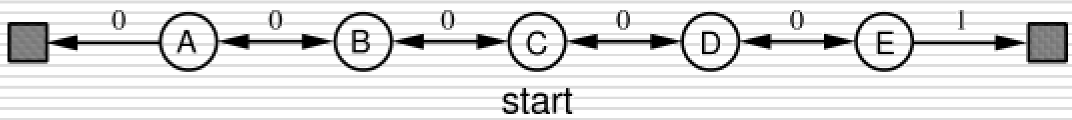
- Пакетное обновление

$$V(s) + \sum_{t=0}^{T-1} \Delta V_t(s)$$

- Свойство сокращения ошибки

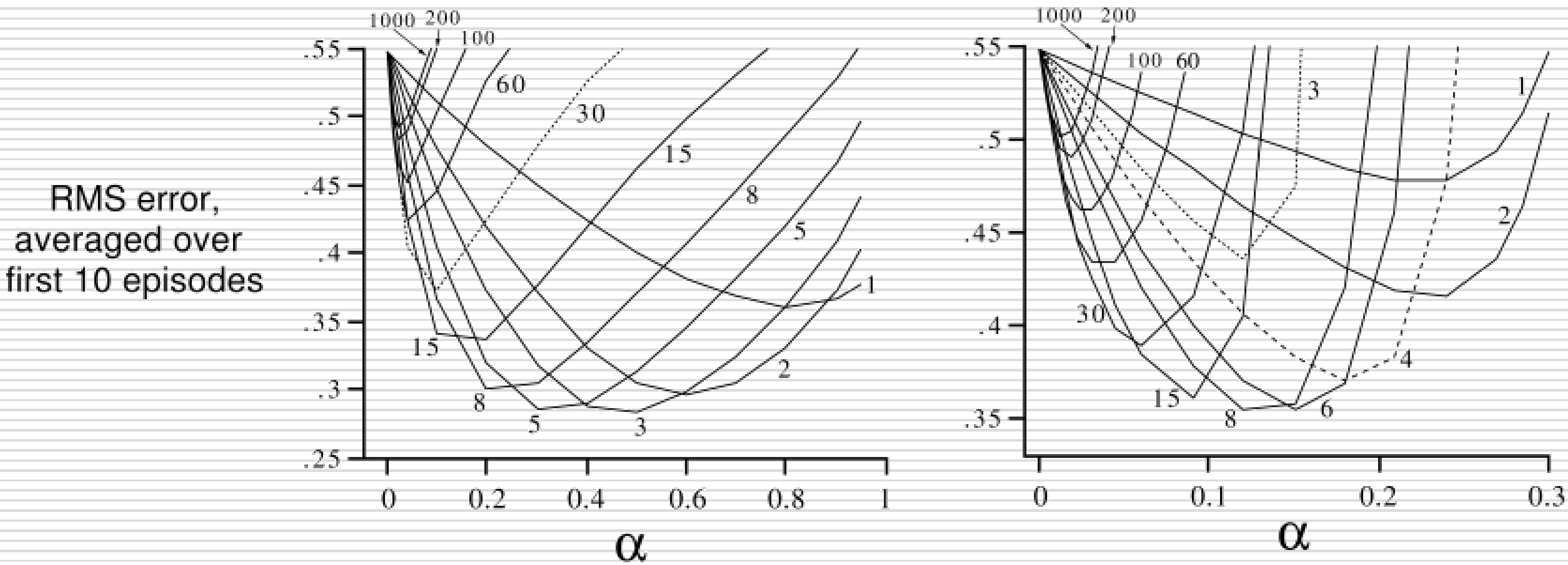
$$\max_s \left| E_\pi\left\{ R_t^{(n)} \;\middle|\; s_t = s \right\} - V^\pi(s) \right| \;\le\; \gamma^n \max_s \left| V(s) - V^\pi(s) \right|.$$

Многошаговые методы временных разностей. Пример.



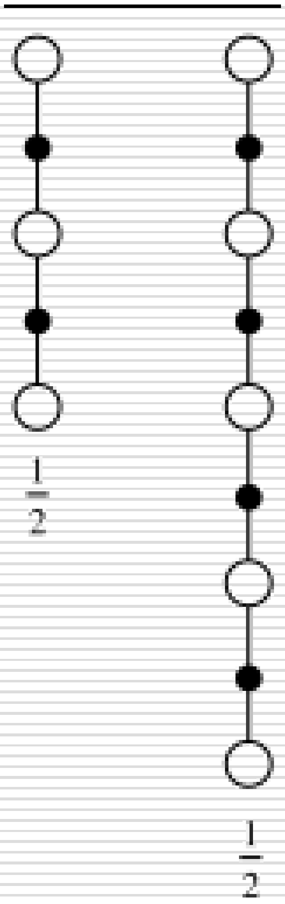
Онлайн обновление

Пакетное обновление



Сложные обновления

$$R_t^{ave} = \frac{1}{2}R_t^{(2)} + \frac{1}{2}R_t^{(4)}$$



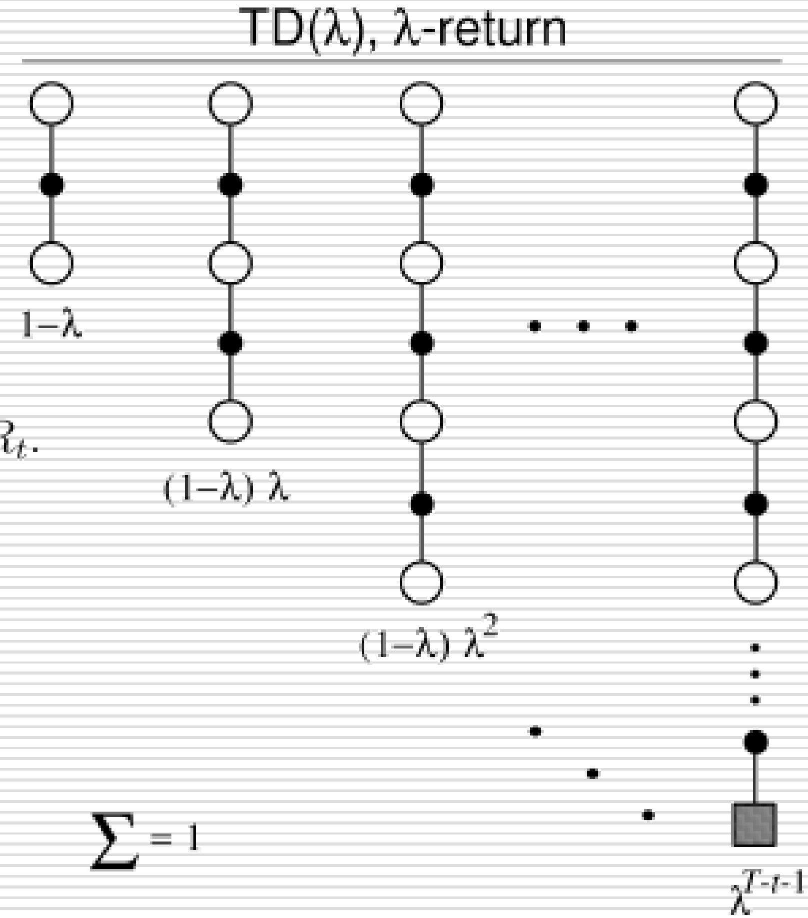
λ-возврат

$$R_t^{\lambda} = (1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}R_t^{(n)}.$$

$$R_t^{\lambda} \;=\; (1-\lambda)\sum_{n=1}^{T-t-1}\lambda^{n-1}R_t^{(n)} \;+\; \lambda^{T-t-1}R_t.$$

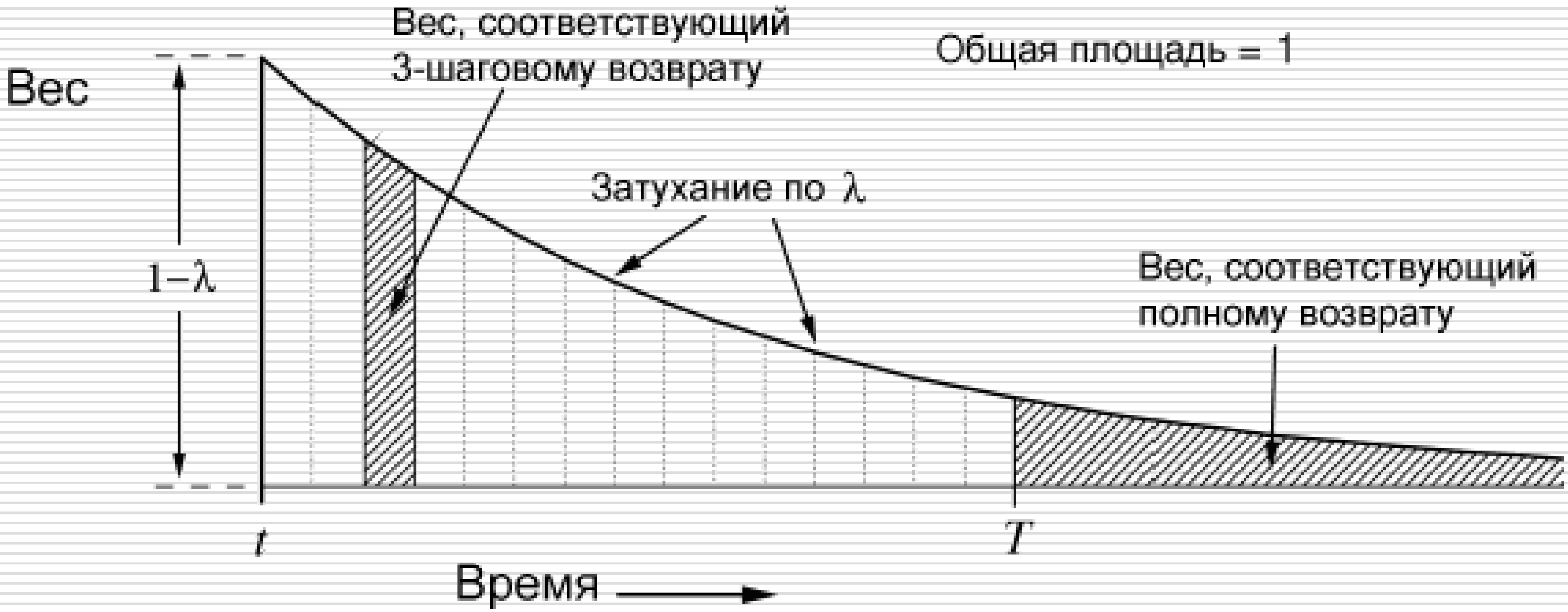
Если λ=1, то получаем $R_t^{\lambda}=R_t$.

Если λ=0, то получаем $R_t^{\lambda}=R_t^{(1)}$.



λ-возврат

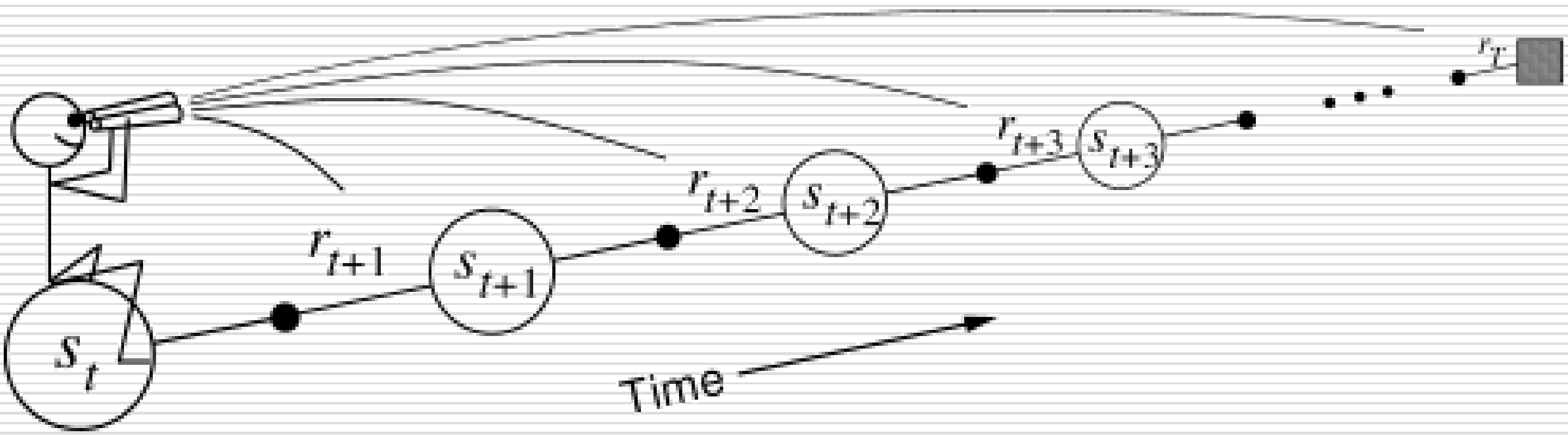
$$R_t^{\lambda} = (1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}R_t^{(n)}.$$



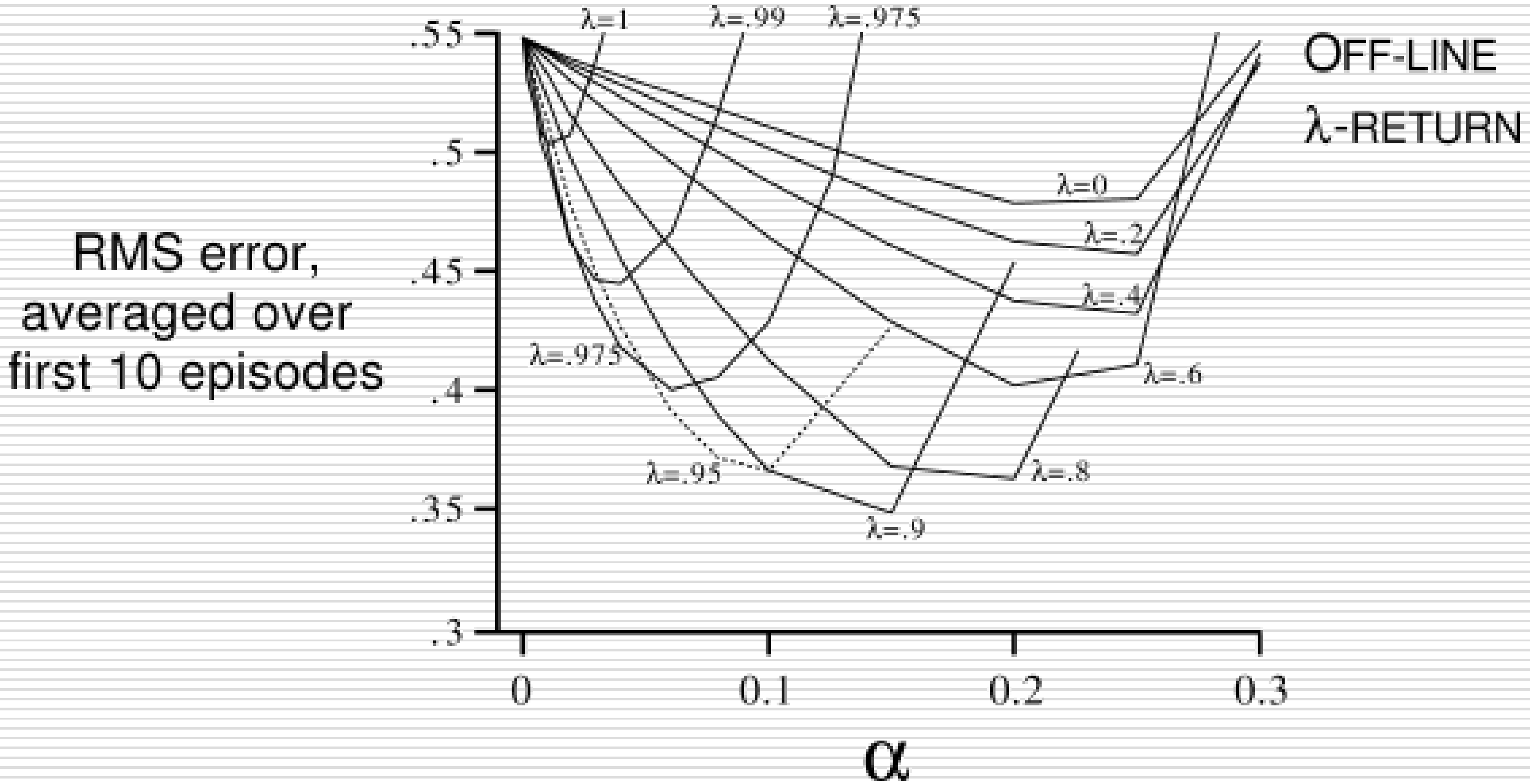
Алгоритм λ-возврата

- Используем обновление

$$\Delta V_t(s_t) = \alpha \left[R_t^\lambda - V_t(s_t) \right].$$



Алгоритм λ-возврата



Следы преемственности

- Вводим для каждого состояния параметр, называемый след преемственности:

$$e_t(s) = \begin{cases} \gamma\lambda e_{t-1}(s) & \text{if } s \neq s_t; \\ \gamma\lambda e_{t-1}(s) + 1 & \text{if } s = s_t, \end{cases}$$

Алгоритм TD(λ)

- След преемственности показывает, «как давно» мы были в состоянии s, и, соответственно, насколько оно ответственно за дальнейшее:

$$e_t(s) = \begin{cases} \gamma\lambda e_{t-1}(s) & \text{if } s \neq s_t; \\ \gamma\lambda e_{t-1}(s) + 1 & \text{if } s = s_t, \end{cases}$$

- «Отвечаем» мы согласно ошибке временных разностей

$$\delta_t = r_{t+1} + \gamma V_t(s_{t+1}) - V_t(s_t).$$

- Получаем следующее обновление

$$\Delta V_t(s) = \alpha\delta_t e_t(s), \qquad \text{for all } s \in \mathcal{S}.$$

Алгоритм TD(λ)

Инициализация:

$V(s) \leftarrow$ произвольно,

$e(s) \leftarrow 0$, для всех $s \in S$.

Повторять (для всех эпизодов)

$s \leftarrow$ начальное состояние

Повторять (для всех шагов эпизода)

$a \leftarrow$ действие для s согласно π.

Выполнить a, узнать r и s'.

$\delta \leftarrow r+\gamma V(s')-V(s)$

$e(s) \leftarrow e(s)+1$

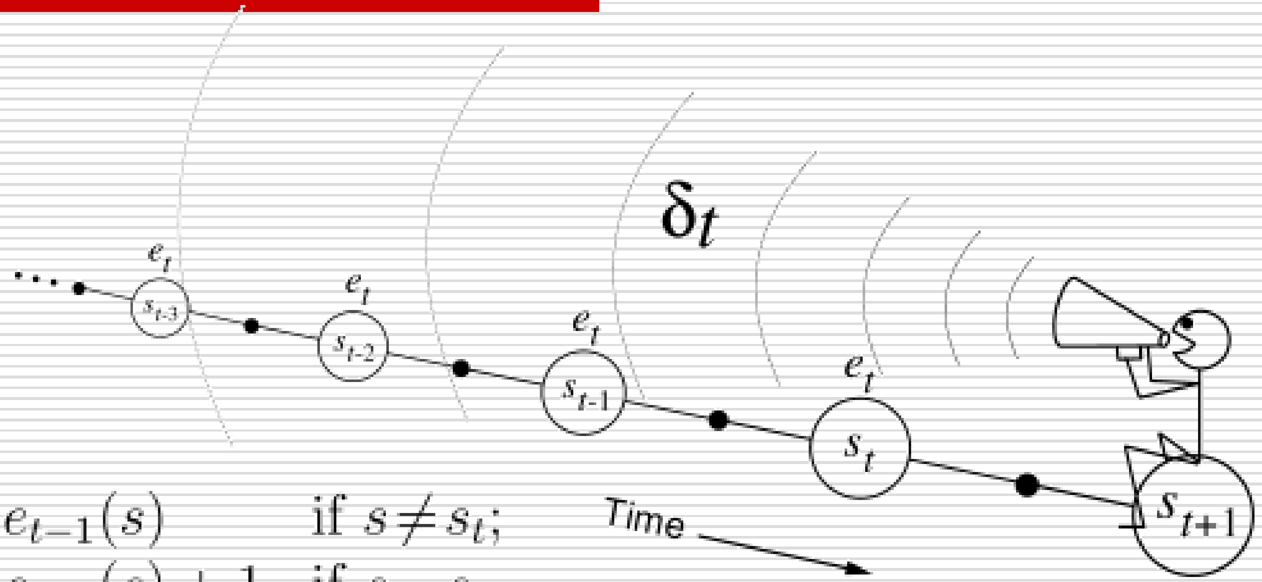
Для всех s:

$V(s) \leftarrow V(s)+\alpha\delta e(s)$

$e(s) \leftarrow \gamma\lambda e(s)$

$s \leftarrow s'$

Алгоритм TD(λ)

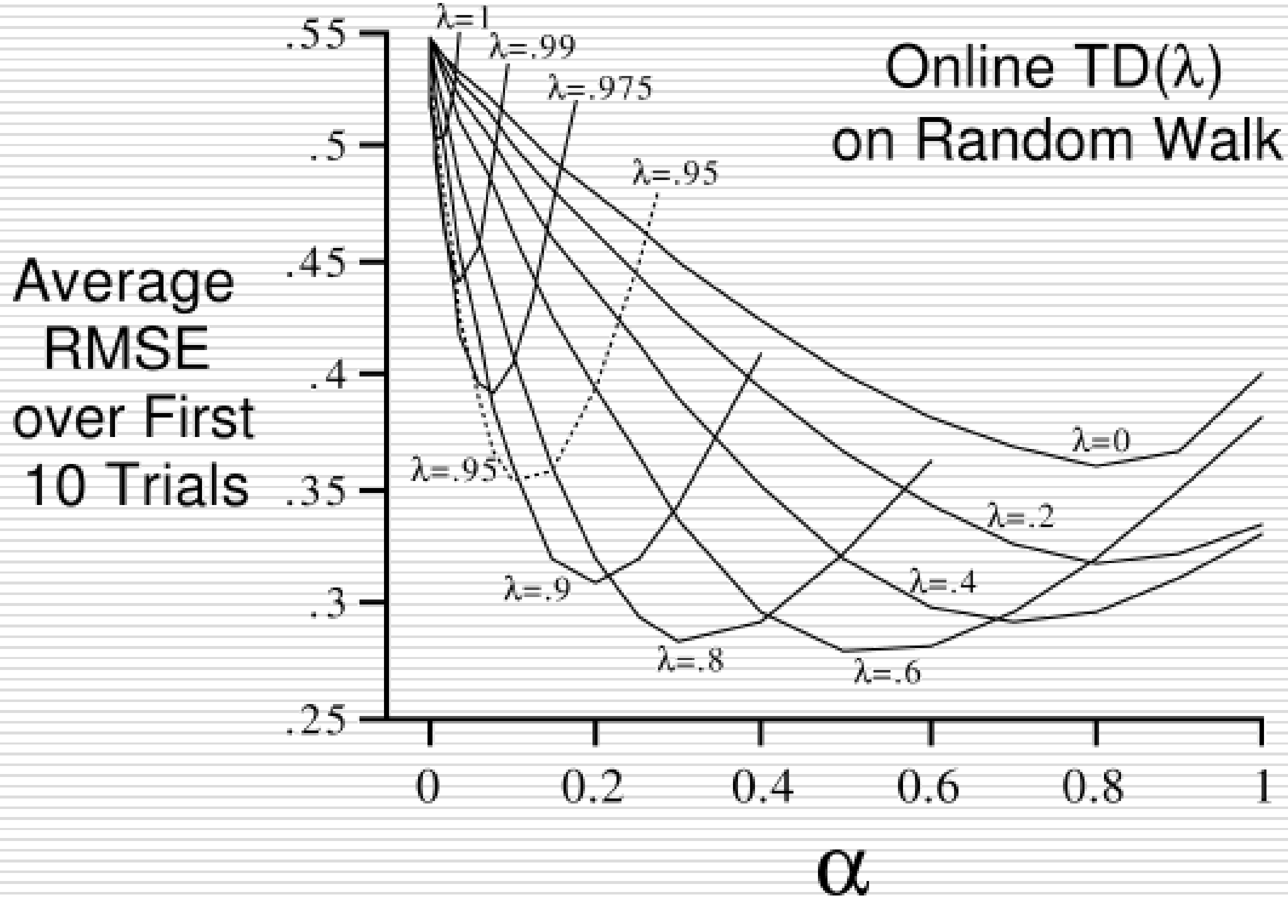


$$e_t(s) = \begin{cases} \gamma\lambda e_{t-1}(s) & \text{if } s \neq s_t; \\ \gamma\lambda e_{t-1}(s) + 1 & \text{if } s = s_t, \end{cases}$$

Если λ=0, то все $e(s)$=0, кроме s=s_t ⇒ получаем эквивалент TD(0)

Если λ=1, то все e затухает только по γ ⇒ получаем эквивалент метода Монте-Карло.

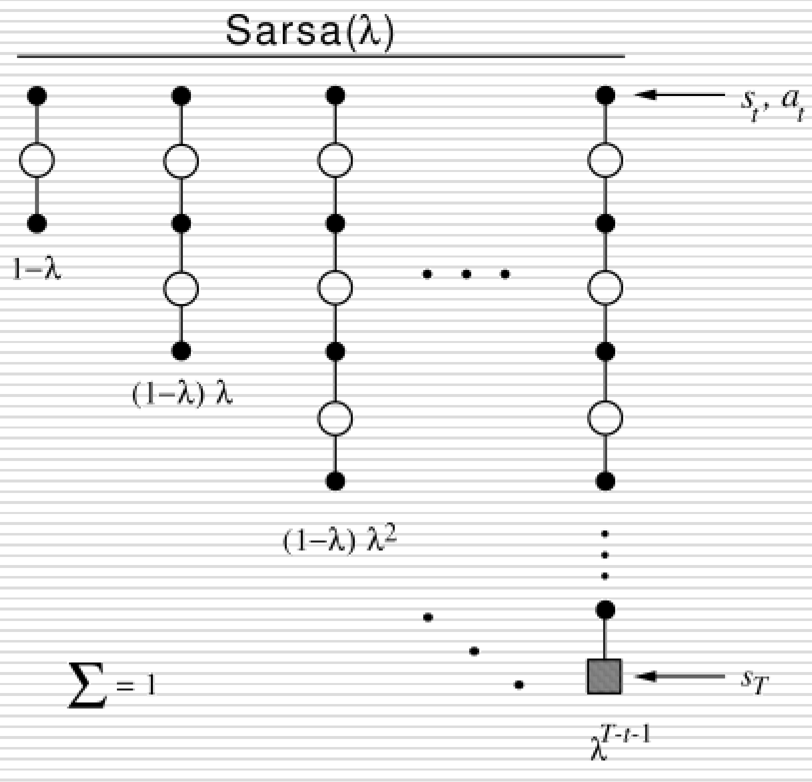
Алгоритм TD(λ). Пример



Sarsa(λ)

- Для управления нам нужно вычислять Q. Введём след преемственности для пары (состояние,действие):

$$e_t(s,a) = \begin{cases} \gamma\lambda e_{t-1}(s,a) + 1 & \text{if } s = s_t \text{ and } a = a_t; \\ \gamma\lambda e_{t-1}(s,a) & \text{otherwise.} \end{cases}$$



- Тогда

$$Q_{t+1}(s,a) = Q_t(s,a) + \alpha\,\delta_t\, e_t(s,a),$$

$$\delta_t = r_{t+1} + \gamma Q_t(s_{t+1}, a_{t+1}) - Q_t(s_t, a_t)$$

Алгоритм Sarsa(λ)

Инициализация:

$Q(s,a) \leftarrow$ произвольно,

$e(s,a) \leftarrow 0$, для всех $s \in S$, $a \in A(s)$.

Повторять (для всех эпизодов)

$(s,a) \leftarrow$ начальное состояние и действие

Повторять (для всех шагов эпизода)

Выполнить a, узнать r и s'.

$a' \leftarrow$ действие для s' согласно ε-жадной по Q стратегии

$\delta \leftarrow r + \gamma Q(s'.a') - Q(s,a)$

$e(s,a) \leftarrow e(s,a) + 1$

Для всех s,a:

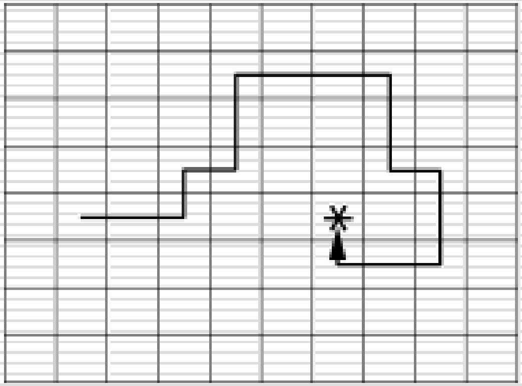
$Q(s,a) \leftarrow Q(s,a) + \alpha\delta e(s,a)$

$e(s,a) \leftarrow \gamma\lambda e(s,a)$

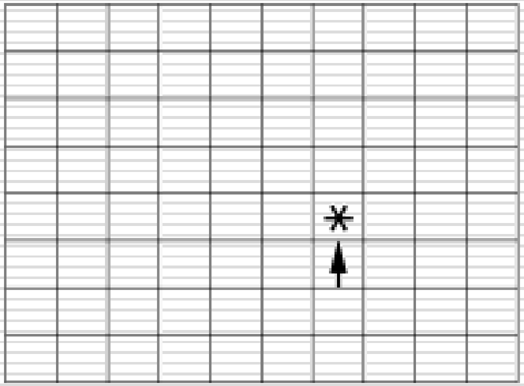
$s \leftarrow s'$, $a \leftarrow a'$

Алгоритм Sarsa(λ)

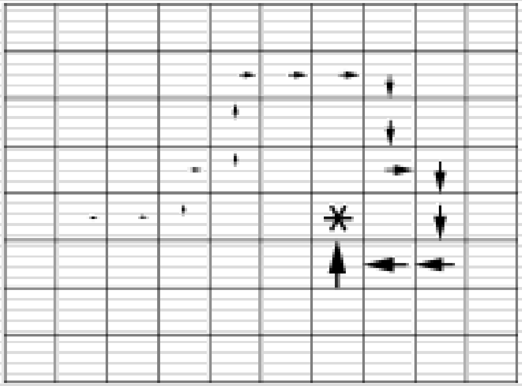
Путь агента



Значение ценности действий, увеличенное одношаговым Sarsa



Значение ценности действий, увеличенное Sarsa(λ), λ=0.9



Следы преемственности и алгоритмы управляющие и оценивающие по разным стратегиям

- Алгоритм Sarsa(λ) неизбежно ограничивается ε-мягкими стратегиями
- Чтобы находить оптимальную стратегию, необходимо оценивать одну стратегию, а управлять по другой. Как совместить Q-learning и следы преемственности?
  - Watkins’s Q(λ)
  - Peng’s Q(λ)

Watkins's Q(λ)

- Допустим, что в момент времени t мы хотим обновить *Q* для пары (s_t, a_t). Пусть два ближайших действия были жадными, а в момент t+3 агент совершит не жадное действие.
- Изучая жадную стратегию, мы тогда можем использовать одношаговый и двухшаговый возврат, но не трех- и более шаговые.

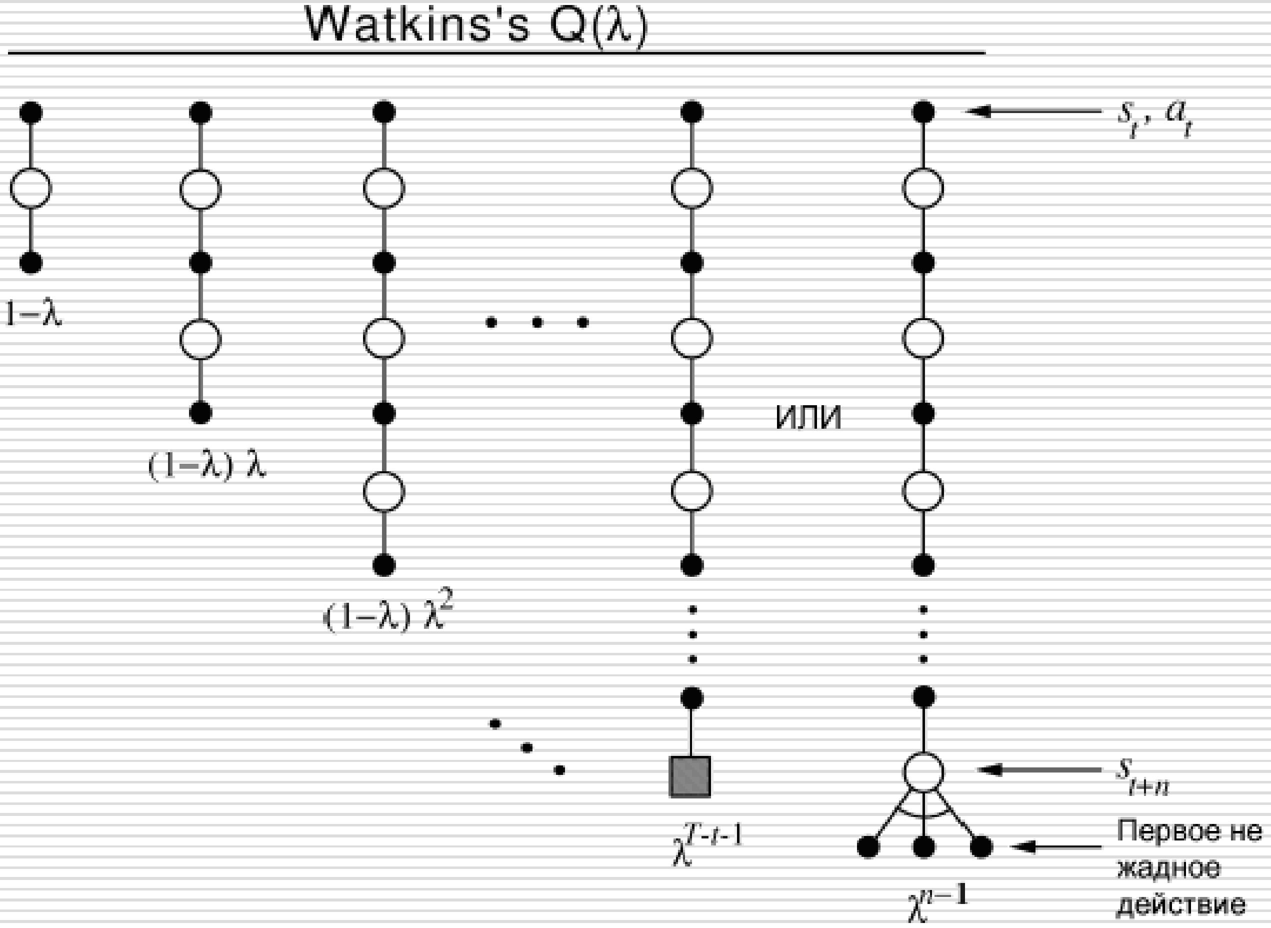
Watkins's Q(λ)

- Допустим, в момент времени t мы хотим обновить Q для пары (s_t, a_t). Пусть два ближайших действия были жадными, а в момент t+3 агент совершит не жадное действие.

- Изучая жадную стратегию, мы тогда можем использовать одношаговый и двухшаговый возврат, но не трех- и более шаговые.

- Алгоритм Watkins's Q(λ) смотрит на один шаг за исследовательское действие, используя знание функции ценности, делая обновление в направлении $r_{t+1} + \gamma \max_a Q_t(s_{t+1}, a)$

Watkins's Q(λ)



Watkins's Q(λ)
s_t, a_t
1−λ
(1−λ) λ
(1−λ) λ²
или
λ^{T-t-1}
s_{t+n}
Первое не
жадное
действие
λ^{n-1}

Алгоритм Watkins's Q(λ)

Инициализация:

$Q(s,a) \leftarrow$ произвольно,

$e(s,a) \leftarrow 0$, для всех $s \in S$, $a \in A(s)$.

Повторять (для всех эпизодов)

$(s,a) \leftarrow$ начальное состояние и действие

Повторять (для всех шагов эпизода)

Выполнить a, узнать r и s'.

$a' \leftarrow$ действие для s' согласно ε-жадной по Q стратегии

$a^* \leftarrow \arg\max_b Q(s',b)$

$\delta \leftarrow r+\gamma Q(s',a^*)-Q(s,a)$

$e(s,a) \leftarrow e(s,a)+1$

Для всех s,a:

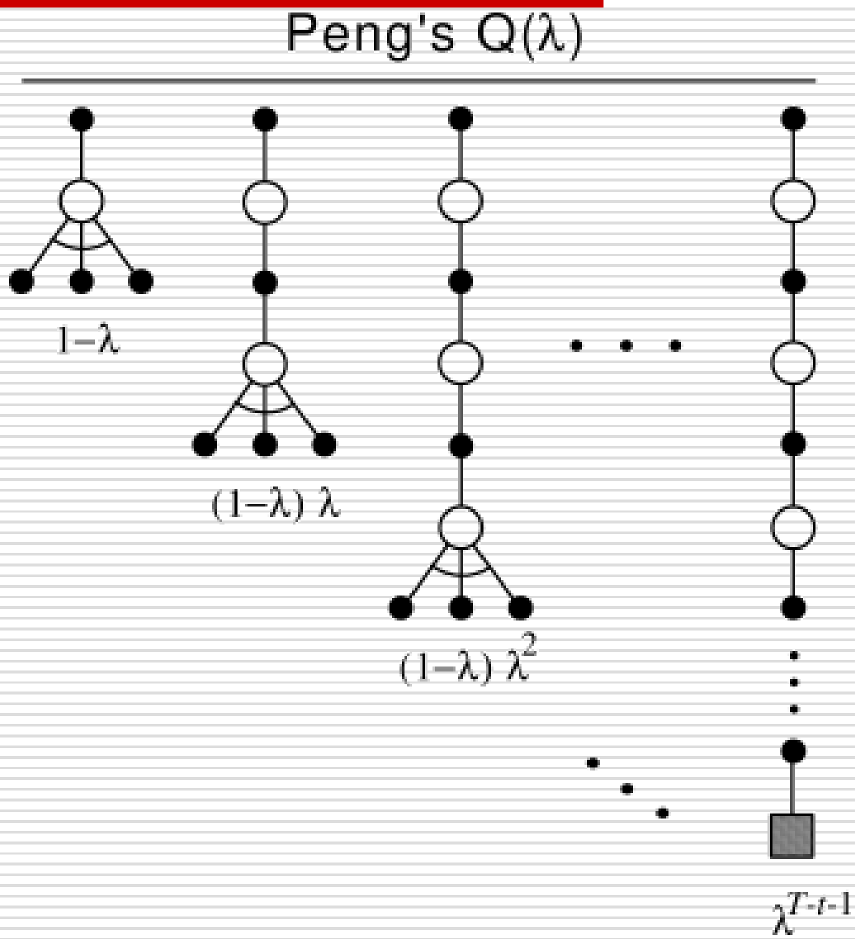
$Q(s,a) \leftarrow Q(s,a)+\alpha\delta e(s,a)$

Если $a'=a^*$ то $e(s,a) \leftarrow \gamma\lambda e(s,a)$

иначе $e(s,a) \leftarrow 0$

$s \leftarrow s'$, $a \leftarrow a'$

Peng’s Q(λ)



Следы преемственности в методах деятеля-критика

- Критик оценивает V, ему нужны следы для каждого состояния. Затем просто используем TD(λ).

- Деятелю нужны следы для пары состояние-действие.
  - Простой метод обновлял предпочтения по правилу

$$p_{t+1}(s,a) = \begin{cases} p_t(s,a) + \alpha\delta_t & \text{if } a = a_t \text{ and } s = s_t \\ p_t(s,a) & \text{otherwise,} \end{cases}$$

  - Используем обновление

$$p_{t+1}(s,a) = p_t(s,a) + \alpha\delta_t e_t(s,a),$$

Следы преемственности в методах деятеля-критика

- Более хитрый деятель действовал по правилу

$$p_{t+1}(s,a) = \begin{cases} p_t(s,a) + \alpha\delta_t[1 - \pi_t(s,a)] & \text{if } a = a_t \text{ and } s = s_t \\ p_t(s,a) & \text{otherwise.} \end{cases}$$

- Для создания эквивалента этого метода используем следующий способ вычисления следа преемственности

$$e_t(s,a) = \begin{cases} \gamma\lambda e_{t-1}(s,a) + 1 - \pi_t(s_t,a_t) & \text{if } s = s_t \text{ and } a = a_t \\ \gamma\lambda e_{t-1}(s,a) & \text{otherwise,} \end{cases}$$

$$p_{t+1}(s,a) = p_t(s,a) + \alpha\delta_t e_t(s,a),$$

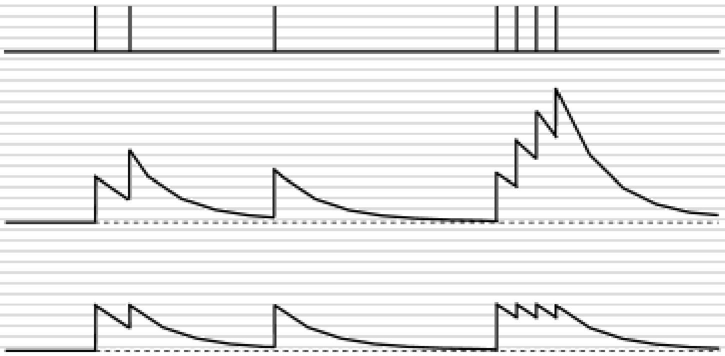
Виды следов преемственности

- Аккумулирующие следы преемственности

$$e_t(s) = \begin{cases} \gamma\lambda e_{t-1}(s) & \text{if } s \neq s_t; \\ \gamma\lambda e_{t-1}(s) + 1 & \text{if } s = s_t, \end{cases}$$

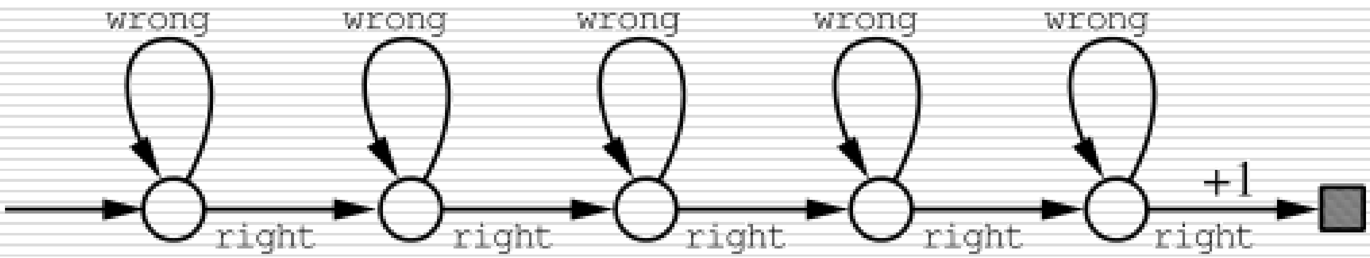
- Замещающие следы преемственности

$$e_t(s) = \begin{cases} \gamma\lambda e_{t-1}(s) & \text{if } s \neq s_t; \\ 1 & \text{if } s = s_t. \end{cases}$$



Виды следов преемственности

- Замещающие следы преемственности для действий



$$e_t(s,a) = \begin{cases} 1 + \gamma\lambda e_{t-1}(s,a) & \text{if } s = s_t \text{ and } a = a_t; \\ 0 & \text{if } s = s_t \text{ and } a \neq a_t; \\ \gamma\lambda e_{t-1}(s,a) & \text{if } s \neq s_t. \end{cases} \qquad \text{for all } s, a$$

Переменные следы преемственности

$$e_t(s) = \begin{cases} \gamma \lambda_t e_{t-1}(s) & \text{if } s \neq s_t; \\ \gamma \lambda_t e_{t-1}(s) + 1 & \text{if } s = s_t, \end{cases}$$

$$\begin{aligned} R_t^\lambda &= \sum_{n=1}^{\infty} R_t^{(n)} (1 - \lambda_{t+n}) \prod_{i=t+1}^{t+n-1} \lambda_i \\ &= \sum_{k=t+1}^{T-1} R_t^{(k-t)} (1 - \lambda_k) \prod_{i=t+1}^{k-1} \lambda_i \; + \; R_t \prod_{i=t+1}^{T-1} \lambda_i \end{aligned}$$

- Например $\lambda_t = \lambda(s_t)$
  - Если мы имеем хорошую оценку для состояния, то можем брать её, игнорируя последующее (λ около 0).
  - Если мы не уверены в оценке состояния, то делая λ близким к 1 мы даём малый вес оценке состояния и большой – тому, что произойдёт позднее.