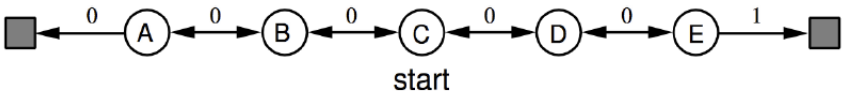
Wprowadzenie do RL 8

Zadanie 1

Dla spaceru przypadkowego omówionego na wykładzie:



Znajdź wartości V(A), V(B), V(C), V(D), V(E):

A. Wykorzystując programowanie dynamiczne (**równanie Bellmana**)
B. Wykorzystując **algorytm TD(0)** (uzupełnij notatnik: RandomWalk_TODO.ipynb)

Przyjmij, że wykorzystana jest **polityka stochastyczna** przy której prawdopodobieństwo ruchu w lewo i w prawo = 0.5.

Zadanie 2

Napisz program implementujący poniższy **algorytm TD(0)** dla środowiska *FrozenLake* w celu wyliczenia polityki π.

Tabular TD(0) for estimating v_π

Input: the policy π to be evaluated
Algorithm parameter: step size $\alpha \in (0, 1]$
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop for each episode:
  Initialize S
  Loop for each step of episode:
    $A \leftarrow$ action given by π for S
    Take action A, observe R, S'
    $V(S) \leftarrow V(S) + \alpha\big[R + \gamma V(S') - V(S)\big]$
    $S \leftarrow S'$
  until S is terminal

Wykorzystaj notatnik: FrozenLake_TD(0)_TODO.ipynb

Zadanie 3

Napisz program implementujący poniższy **algorytm** dla środowiska *FrozenLake*.

Sarsa (on-policy TD control) for estimating $Q \approx q_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal, \cdot) = 0$

Loop for each episode:
  Initialize S
  Choose A from S using policy derived from Q (e.g., ε-greedy)
  Loop for each step of episode:
    Take action A, observe R, S'
    Choose A' from S' using policy derived from Q (e.g., ε-greedy)
    $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma Q(S', A') - Q(S, A)\big]$
    $S \leftarrow S'$; $A \leftarrow A'$;
  until S is terminal

Wykorzystaj notatnik: **FrozenLake_SARSA_TODO.ipynb**

Zadanie 4

Napisz program implementujący poniższy **algorytm** dla środowiska *FrozenLake*.

Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal, \cdot) = 0$

Loop for each episode:
  Initialize S
  Loop for each step of episode:
    Choose A from S using policy derived from Q (e.g., ε-greedy)
    Take action A, observe R, S'
    $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma \max_a Q(S', a) - Q(S, A)\big]$
    $S \leftarrow S'$
  until S is terminal