

Reinforcement Learning (AI644)

Prepared by Dr. Ashraf Elsayed

Lecture 7



Reinforcement Learning

Lecture 7

Prepared by
Dr. Ashraf Elsayed

Temporal Difference Learning Methods for Prediction

The objectives of this lecture is to:

- Define temporal-difference learning
- Define the temporal-difference error
- Understand the TD(0) algorithm
- Understand the benefits of learning online with TD
- Identify key advantages of TD methods over Dynamic Programming and Monte Carlo methods

What is Temporal Difference (TD) learning?

- Temporal difference (TD) learning refers to a class of model-free reinforcement learning methods which learn by bootstrapping from the current estimate of the value function.
- These methods sample from the environment, like Monte Carlo methods, and perform updates based on current estimates, like dynamic programming methods.
- That means this algorithm can form a Monte Carlo estimate without saving lists of returns.
- In this case we want to be able to learn incrementally before the end of the episode.

TD Prediction

$$v_{\pi}(s) \doteq \mathbb{E}_{\pi}[G_t | S_t = s]$$

$$V(S_t) \leftarrow V(S_t) + \alpha [G_t - V(S_t)]$$

Incremental update rule (as in the bandit)

$$V(S_t) \leftarrow V(S_t) + \alpha \boxed{G_t} - V(S_t)]$$

In MC we need to take samples of full trajectories to compute G_t

Bootstrapping

$$\begin{aligned} G_t &\doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots \\ &= R_{t+1} + \gamma G_{t+1} \end{aligned}$$

$$\begin{aligned} v_\pi(s) &\doteq \mathbb{E}_\pi[G_t | S_t = s] \\ &= \mathbb{E}_\pi[R_{t+1} + \gamma \boxed{G_{t+1}} | S_t = s] \\ &= R_{t+1} + \gamma v_\pi(S_{t+1}) \end{aligned}$$

TD Prediction

We need to take samples of full trajectories to compute G_t

$$v_{\pi}(s) \doteq \mathbb{E}_{\pi}[G_t | S_t = s]$$

$$V(S_t) \leftarrow V(S_t) + \alpha [G_t - V(S_t)]$$

$$V(S_t) \leftarrow V(S_t) + \alpha [G_t - V(S_t)]$$

$$G_t \approx R_{t+1} + \gamma V(S_{t+1})$$

$$V(S_t) \leftarrow V(S_t) + \alpha [R_{t+1} + \gamma V(S_{t+1}) - V(S_t)]$$

$$\delta_t = R_{t+1} + \gamma V(S_{t+1}) - V(S_t)$$

TD Error

In Dynamic Programming

$$V(S_t) \leftarrow V(S_t) + \alpha [R_{t+1} + \gamma V(S_{t+1}) - V(S_t)]$$

Iterative Policy Evaluation, for estimating $V \approx v_\pi$

Input π , the policy to be evaluated

$V \leftarrow \vec{0}, V' \leftarrow \vec{0}$

Loop:

$\Delta \leftarrow 0$

Loop for each $s \in \mathcal{S}$:

→ $V'(s) \leftarrow \sum_a \pi(a | s) \sum_{s', r} p(s', r | s, a) [r + \gamma V(s')]$

$\Delta \leftarrow \max(\Delta, |V'(s) - V(s)|)$

$V \leftarrow V'$

until $\Delta < \theta$ (a small positive number)

Output $V \approx v_\pi$

1-Step TD

$$V(S_t) \leftarrow V(S_t) + \alpha [R_{t+1} + \gamma V(\boxed{S_{t+1}}) - V(S_t)]$$

$$S_t \leftarrow S_{t+1}$$

S, A, R, S, A, R, S, A, R, S, A, R, ...

$$V(S_t) \leftarrow V(S_t) + \alpha [R_{t+1} + \gamma V(S_{t+1}) - V(S_t)]$$

$$S_t \leftarrow S_{t+1}$$

update

S, A, R, S, A, R, S, A, R, S, A, R, ...

TD(o) Algorithm

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 $V(\text{terminal}) = 0$

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'

$V(S) \leftarrow V(S) + \alpha[R + \gamma V(S') - V(S)]$

$S \leftarrow S'$

 until S is terminal

Example: Driving Home

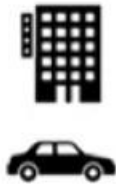
Driving Home Each day as you drive home from work, you try to predict how long it will take to get home. When you leave your office, you note the time, the day of week, the weather, and anything else that might be relevant. Say on this Friday you are leaving at exactly 6 o'clock, and you estimate that it will take 30 minutes to get home. As you reach your car it is 6:05, and you notice it is starting to rain. Traffic is often slower in the rain, so you reestimate that it will take 35 minutes from then, or a total of 40 minutes. Fifteen minutes later you have completed the highway portion of your journey in good time. As you exit onto a secondary road you cut your estimate of total travel time to 35 minutes. Unfortunately, at this point you get stuck behind a slow truck, and the road is too narrow to pass. You end up having to follow the truck until you turn onto the side street where you live at 6:40. Three minutes later you are home. The sequence of states, times, and predictions is thus as follows:

Example: Driving Home

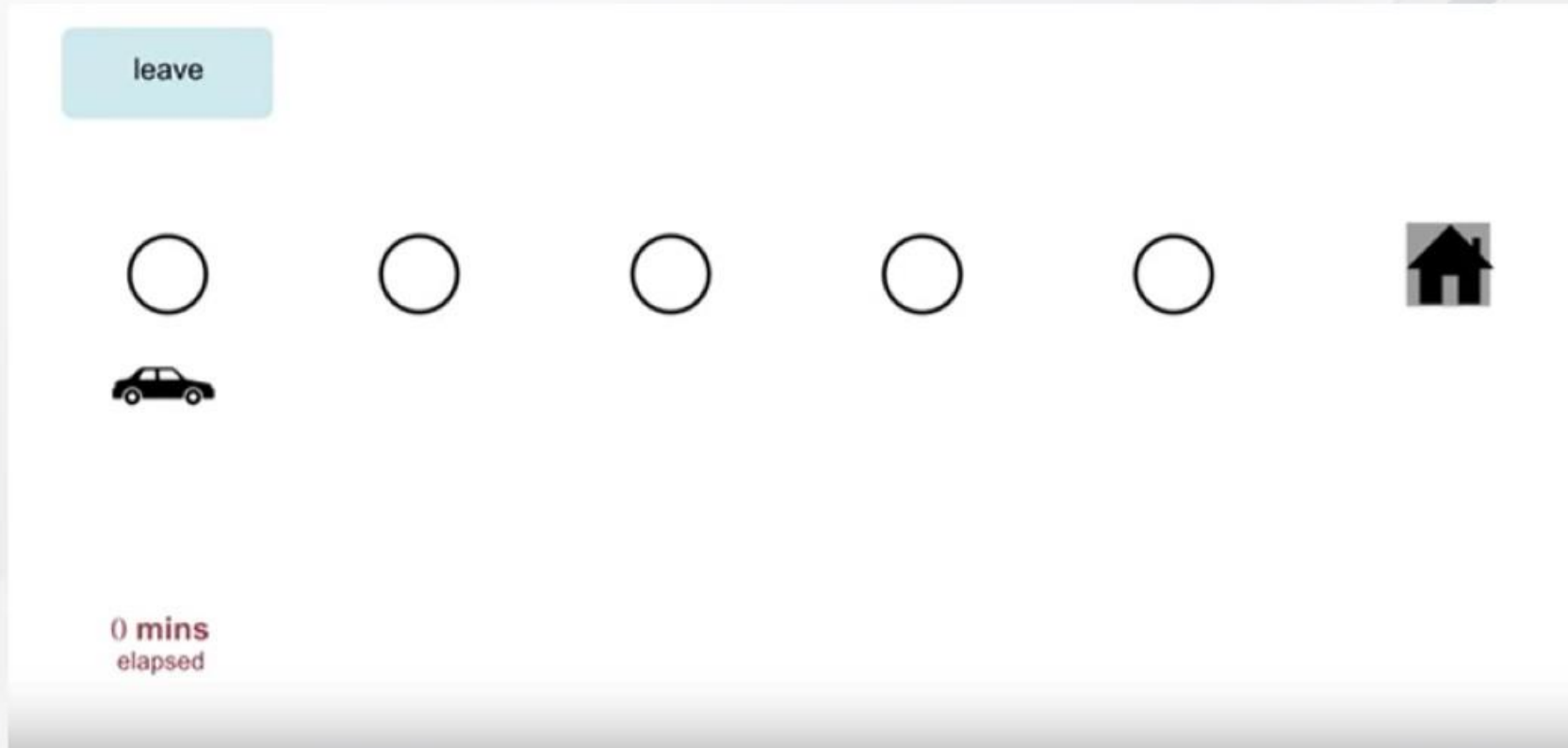
<i>State</i>	<i>Elapsed Time (minutes)</i>	<i>Predicted Time to Go</i>	<i>Predicted Total Time</i>
leaving office, friday at 6	0	30	30
reach car, raining	5	35	40
exiting highway	20	15	35
2ndary road, behind truck	30	10	40
entering home street	40	3	43
arrive home	43	0	43

Example: Driving Home

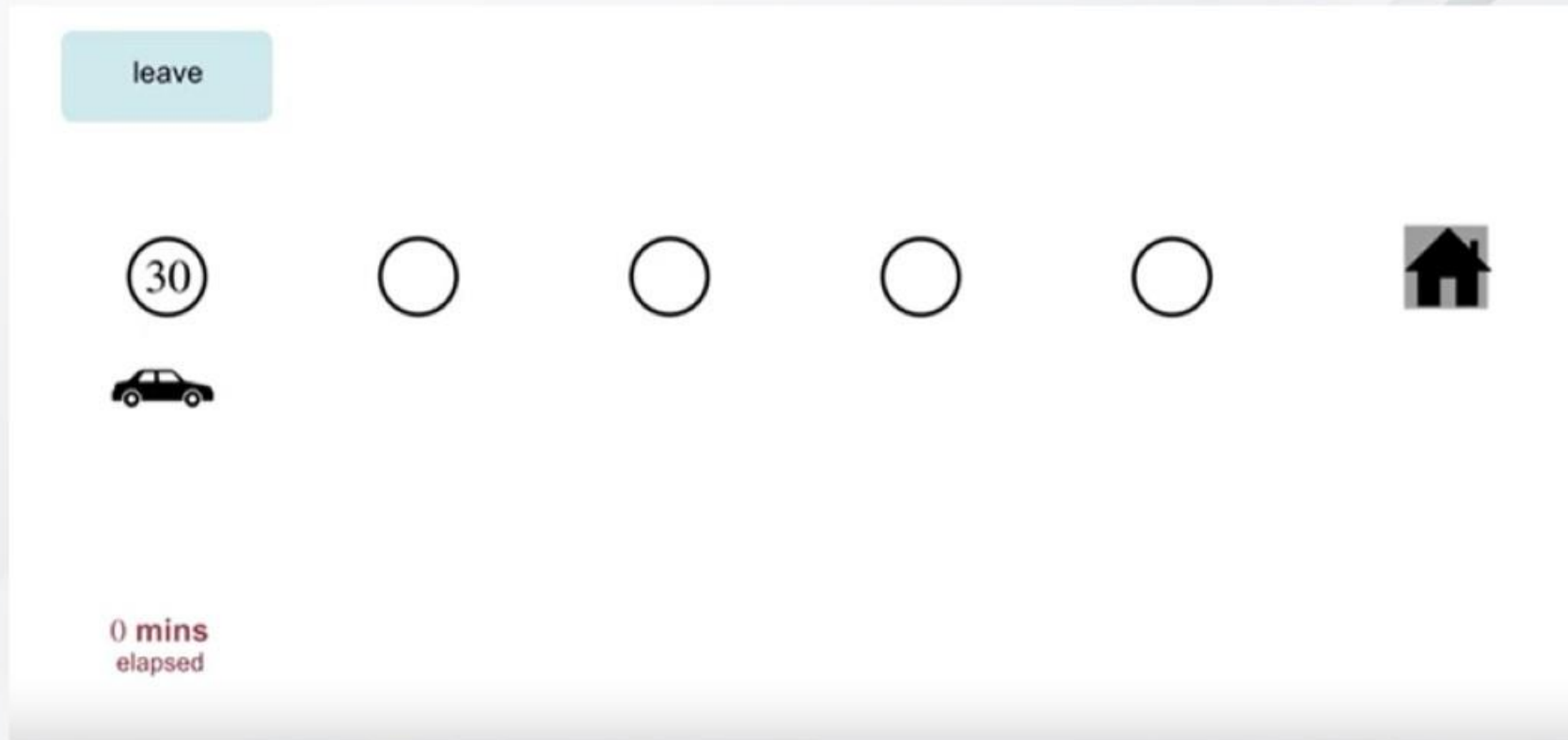
Driving Home



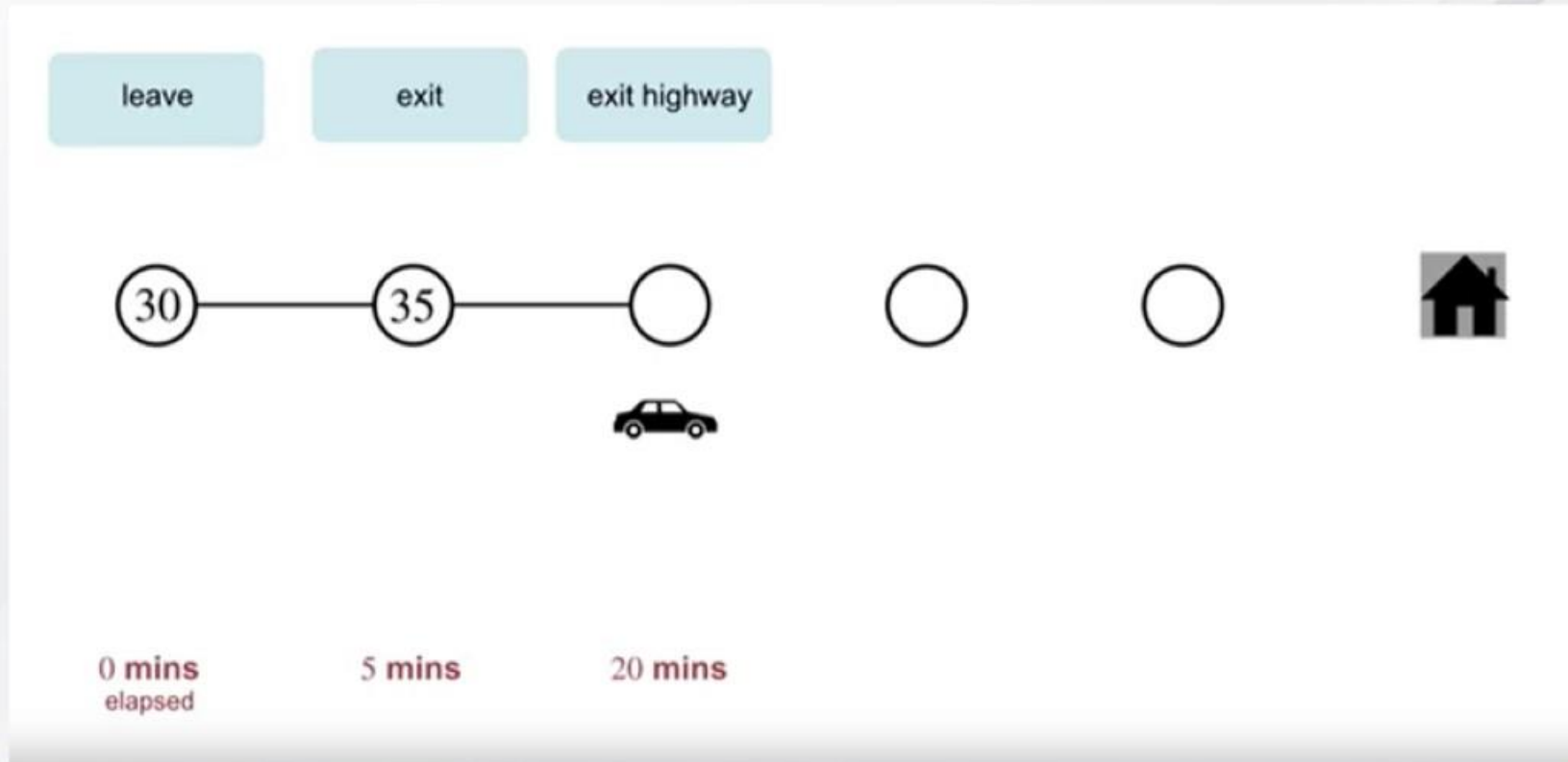
Example: Driving Home



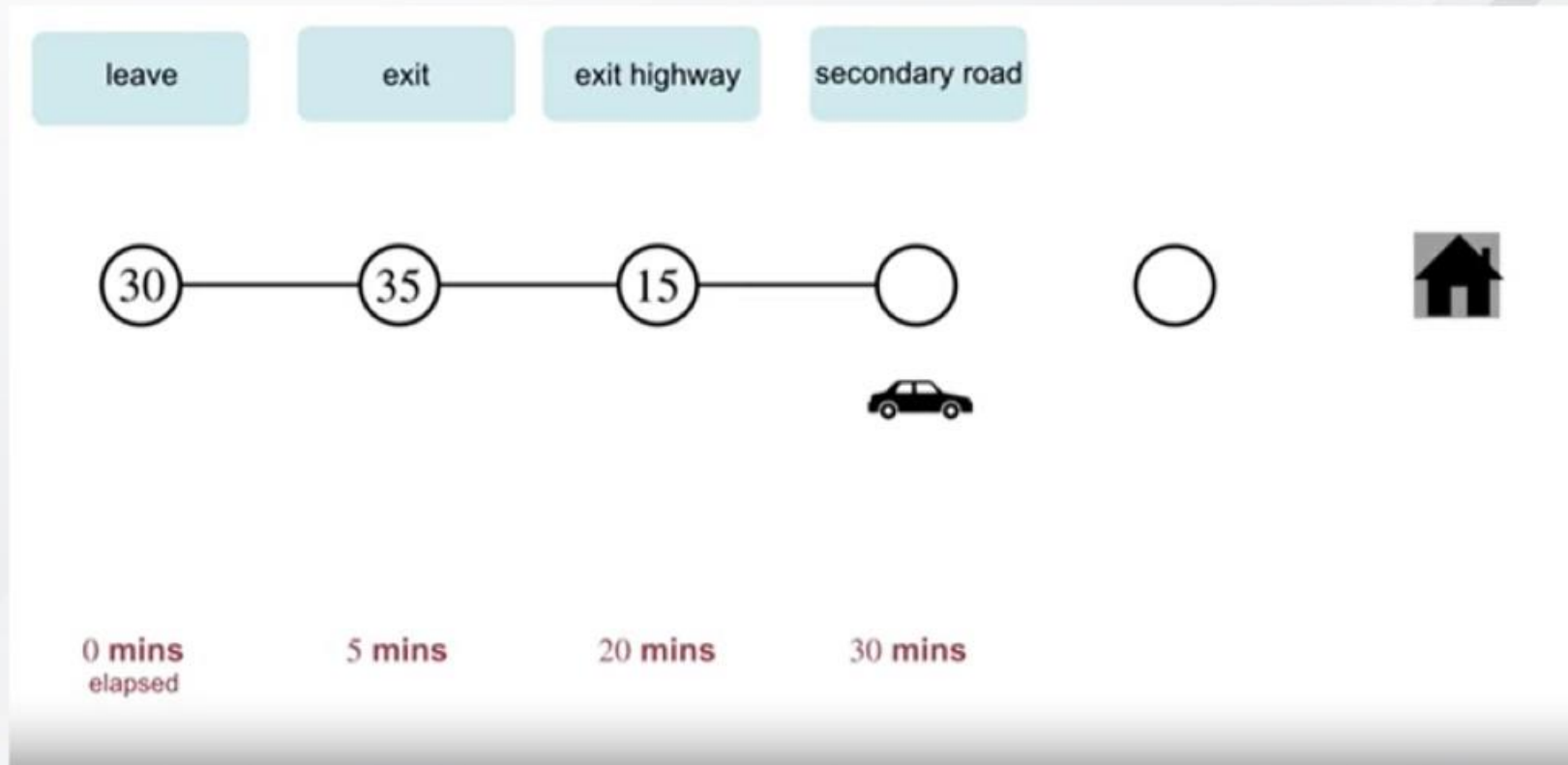
Example: Driving Home



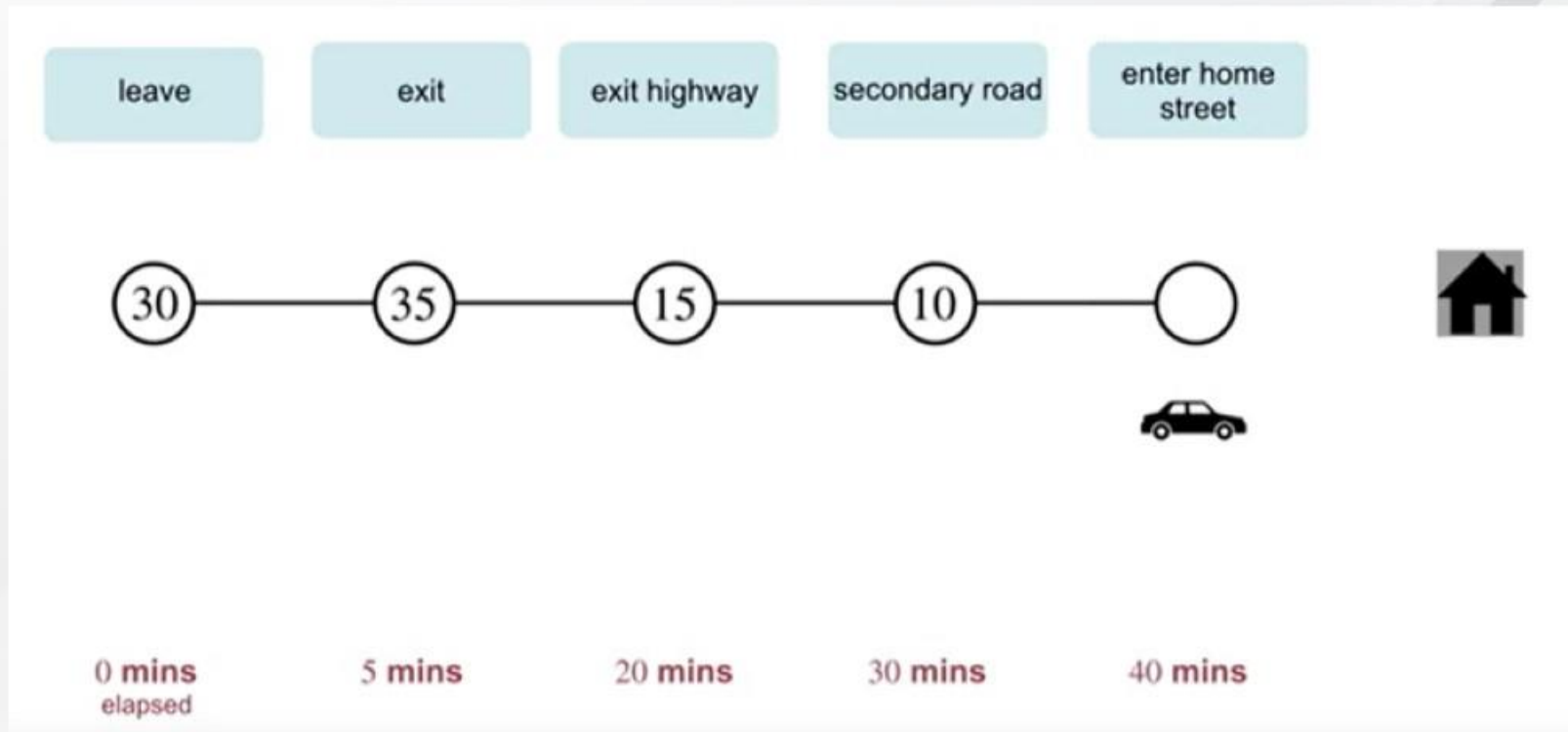
Example: Driving Home



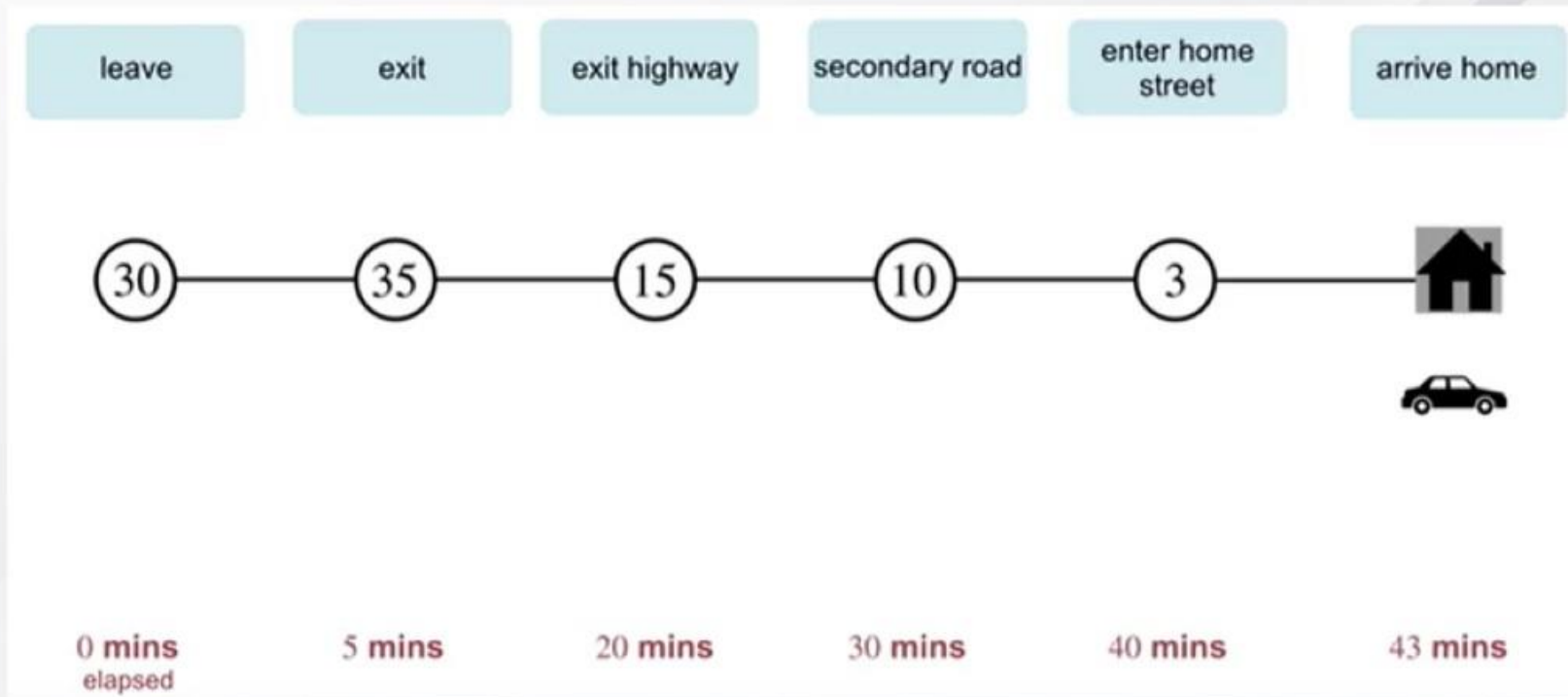
Example: Driving Home



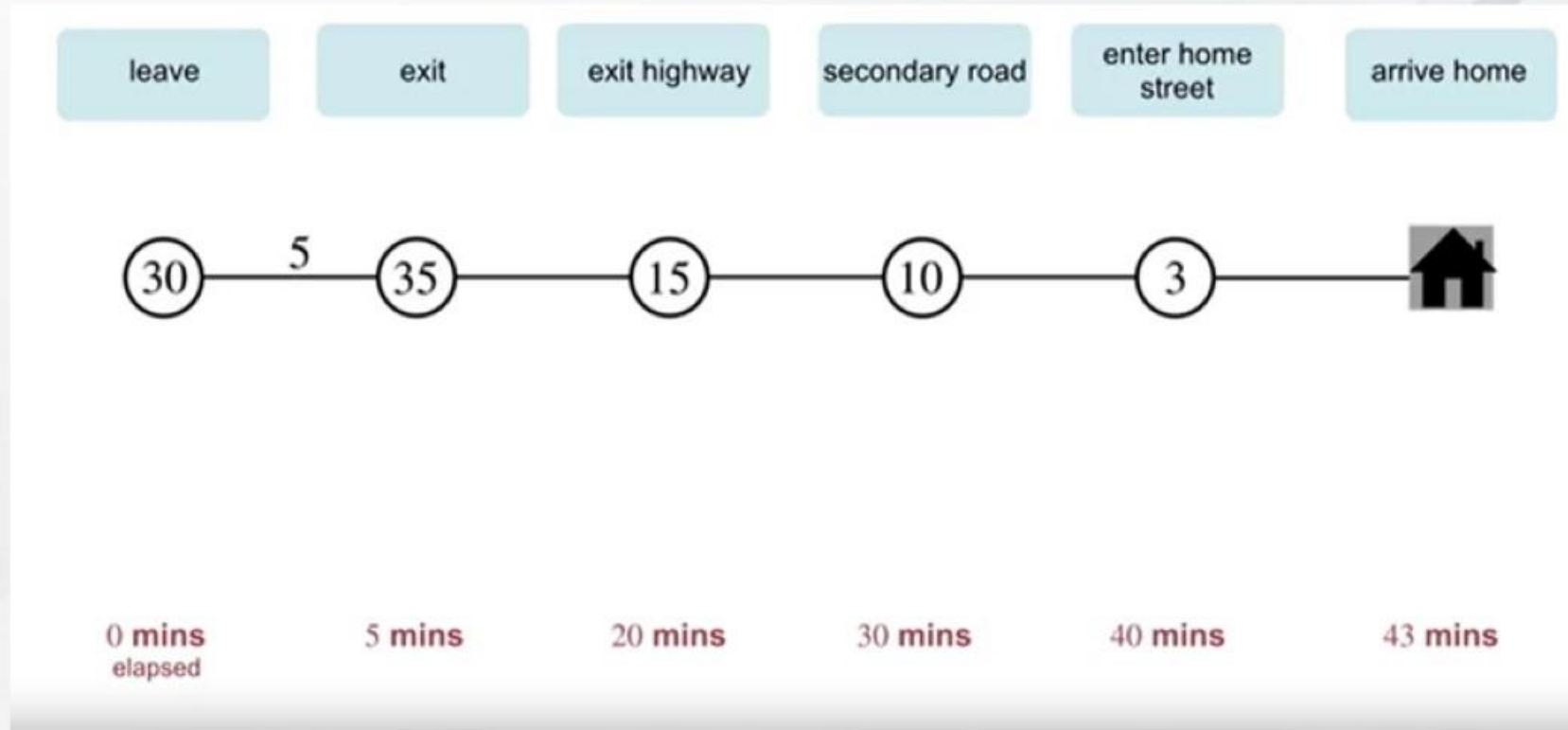
Example: Driving Home



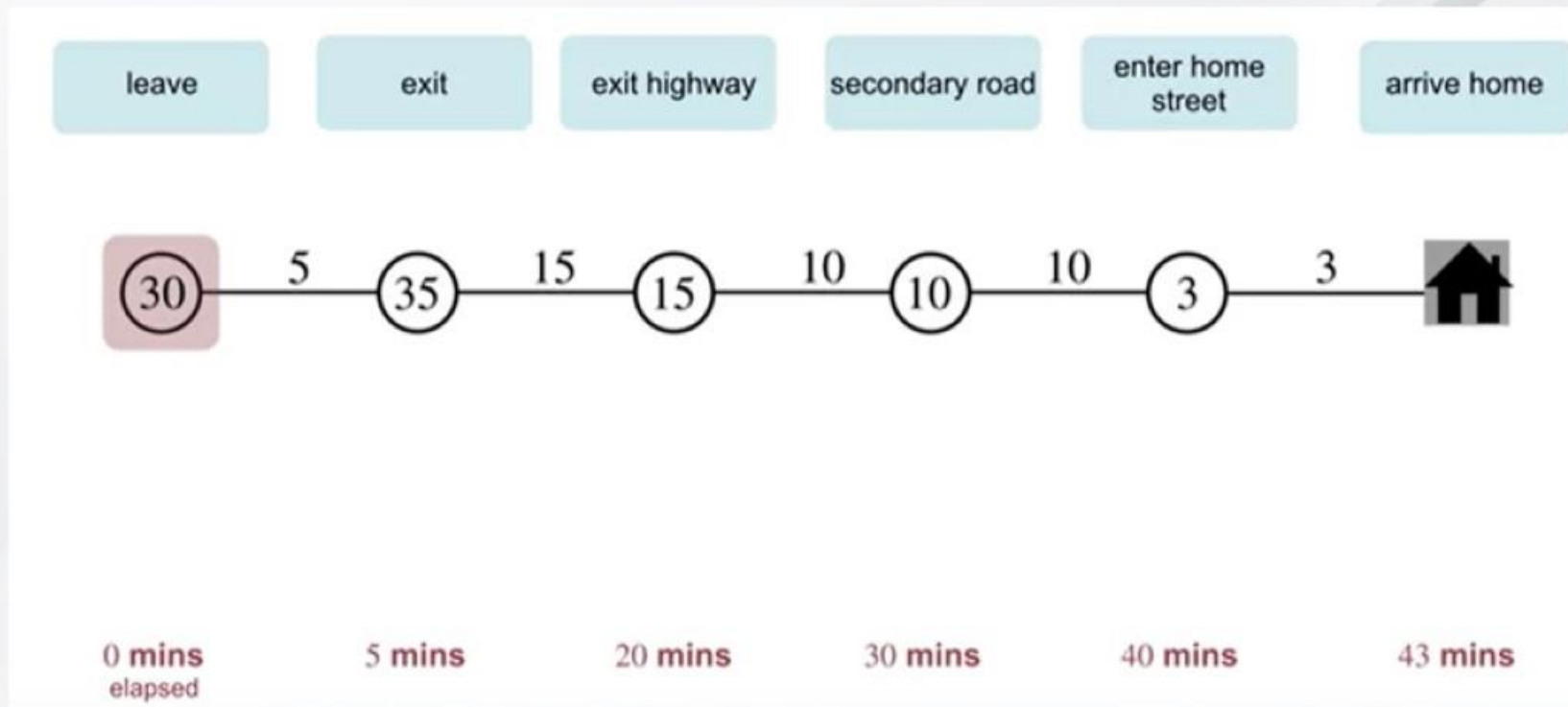
Example: Driving Home



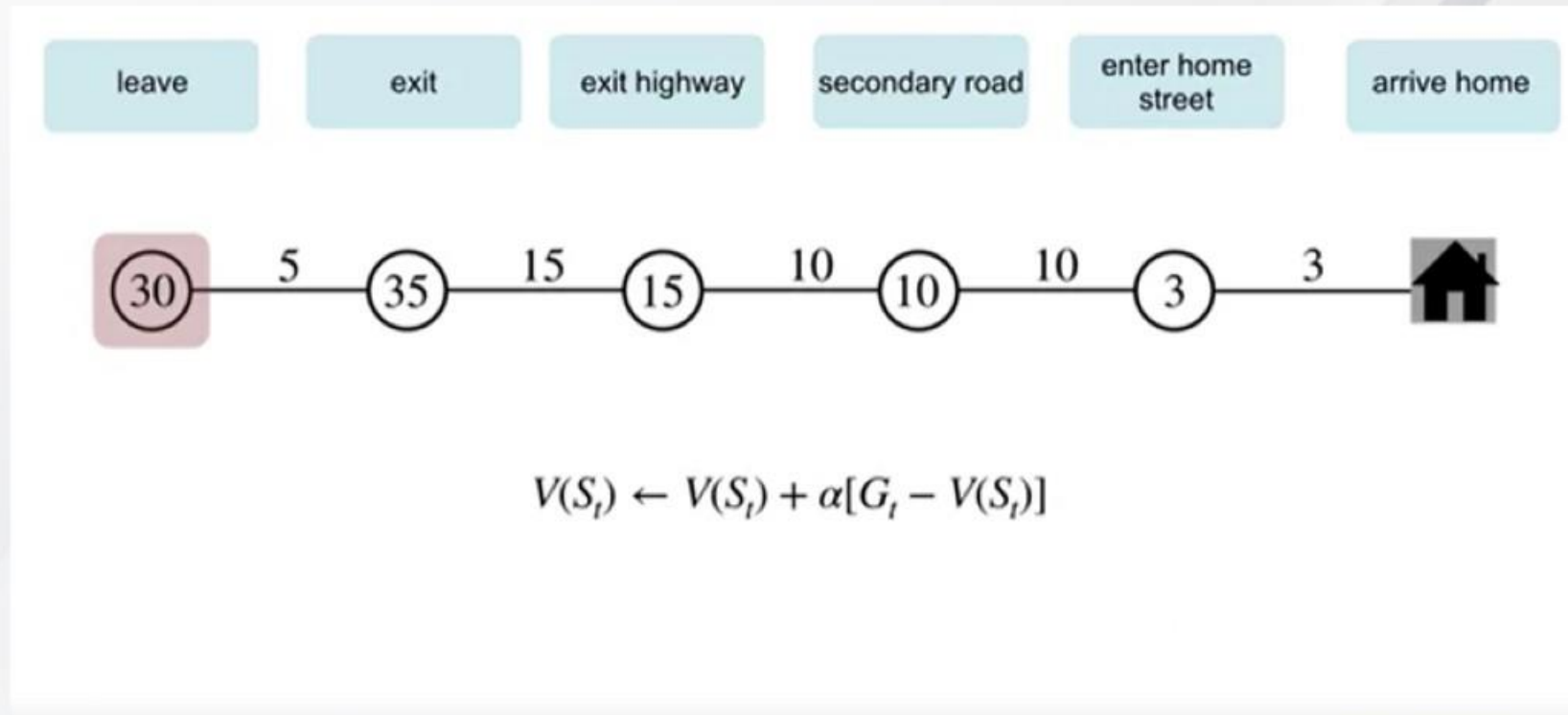
Example: Driving Home



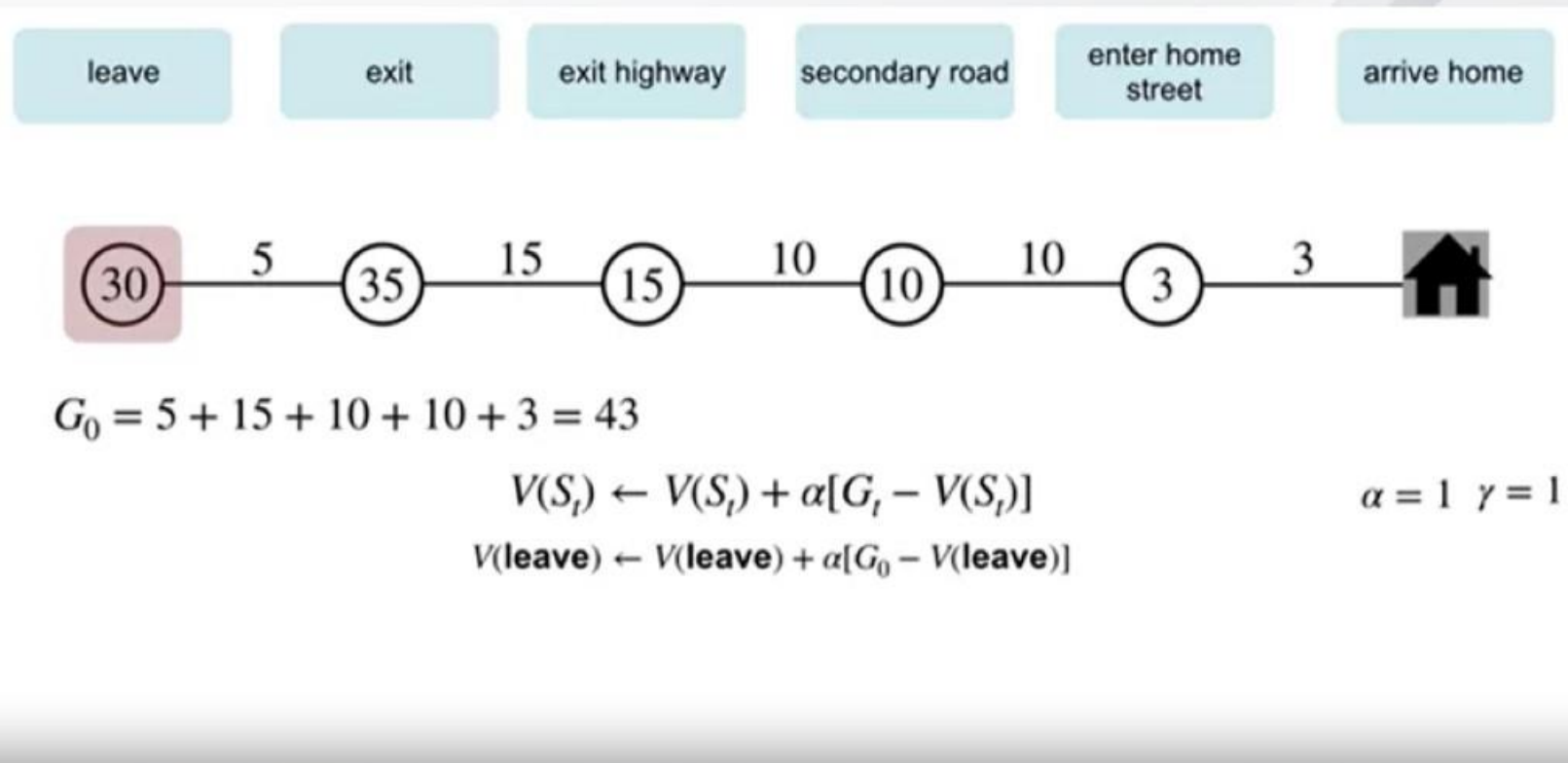
Example: Driving Home



Example: Driving Home (Using MC)



Example: Driving Home (Using MC)



Example: Driving Home (Using MC)

leave

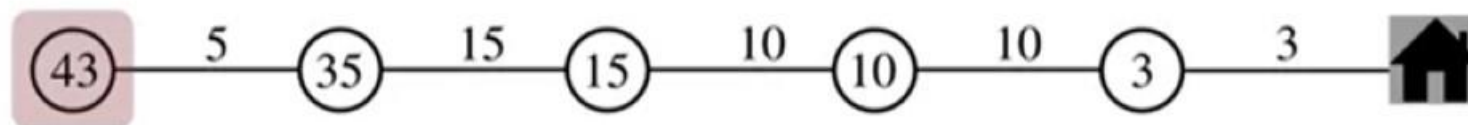
exit

exit highway

secondary road

enter home
street

arrive home



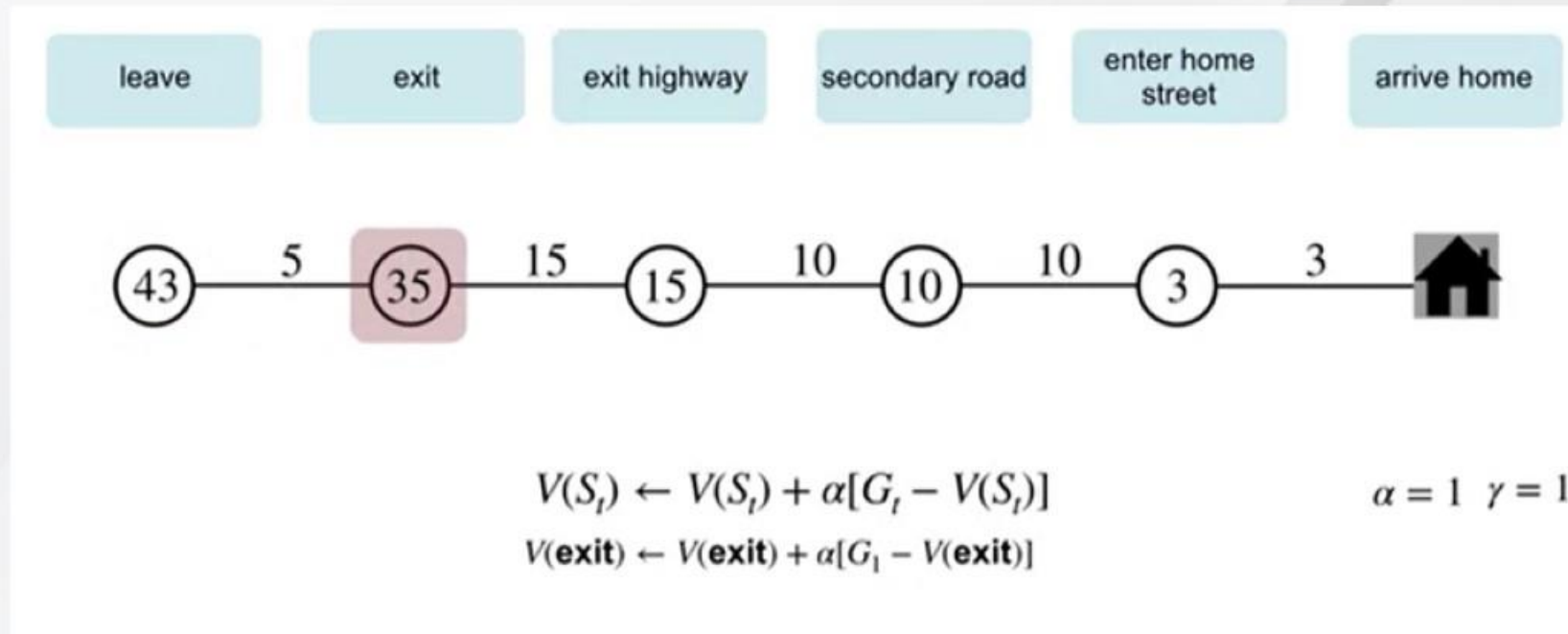
$$G_0 = 5 + 15 + 10 + 10 + 3 = 43$$

$$V(S_t) \leftarrow V(S_t) + \alpha[G_t - V(S_t)]$$

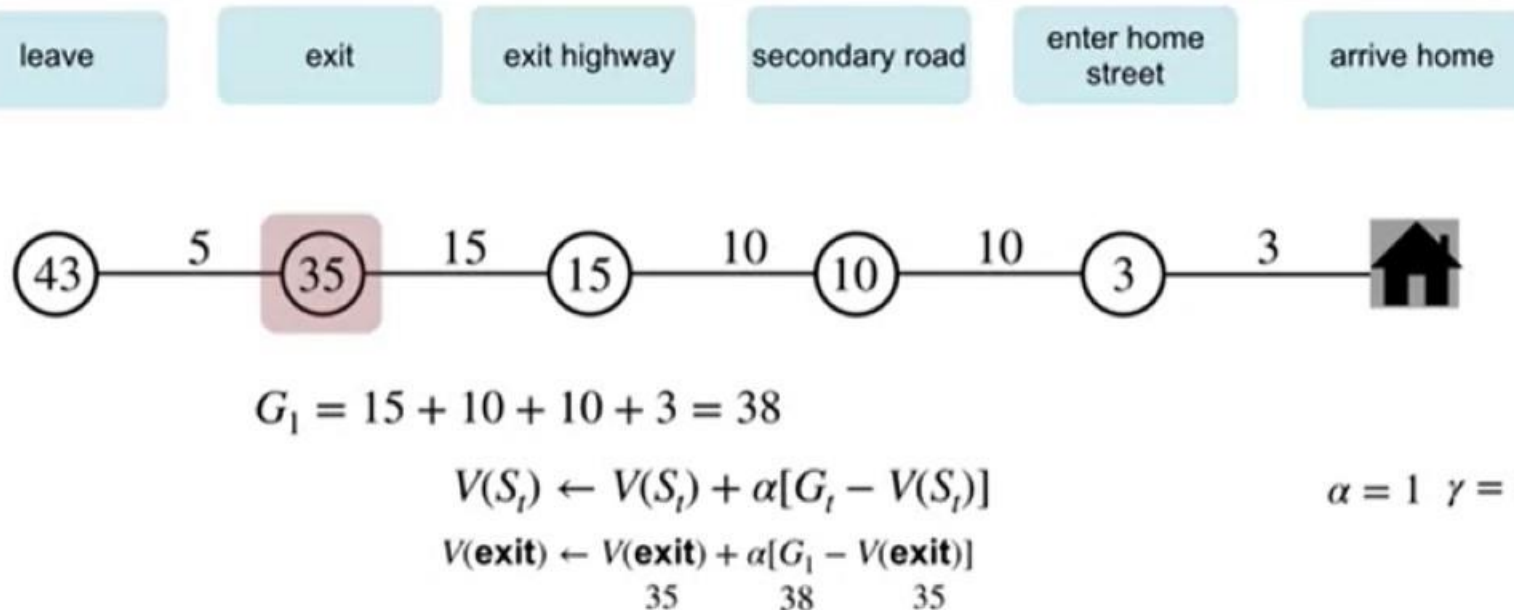
$$\alpha = 1 \quad \gamma = 1$$

$$V(\text{leave}) \leftarrow \underset{30}{V(\text{leave})} + \underset{43}{\alpha[G_0 - \underset{30}{V(\text{leave})}]}$$

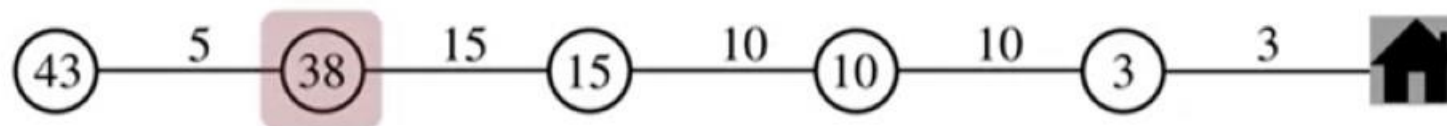
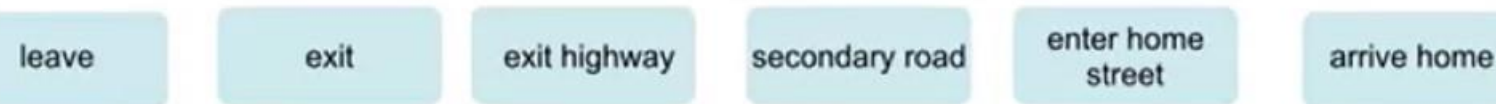
Example: Driving Home (Using MC)



Example: Driving Home (Using MC)



Example: Driving Home (Using MC)



$$G_1 = 15 + 10 + 10 + 3 = 38$$

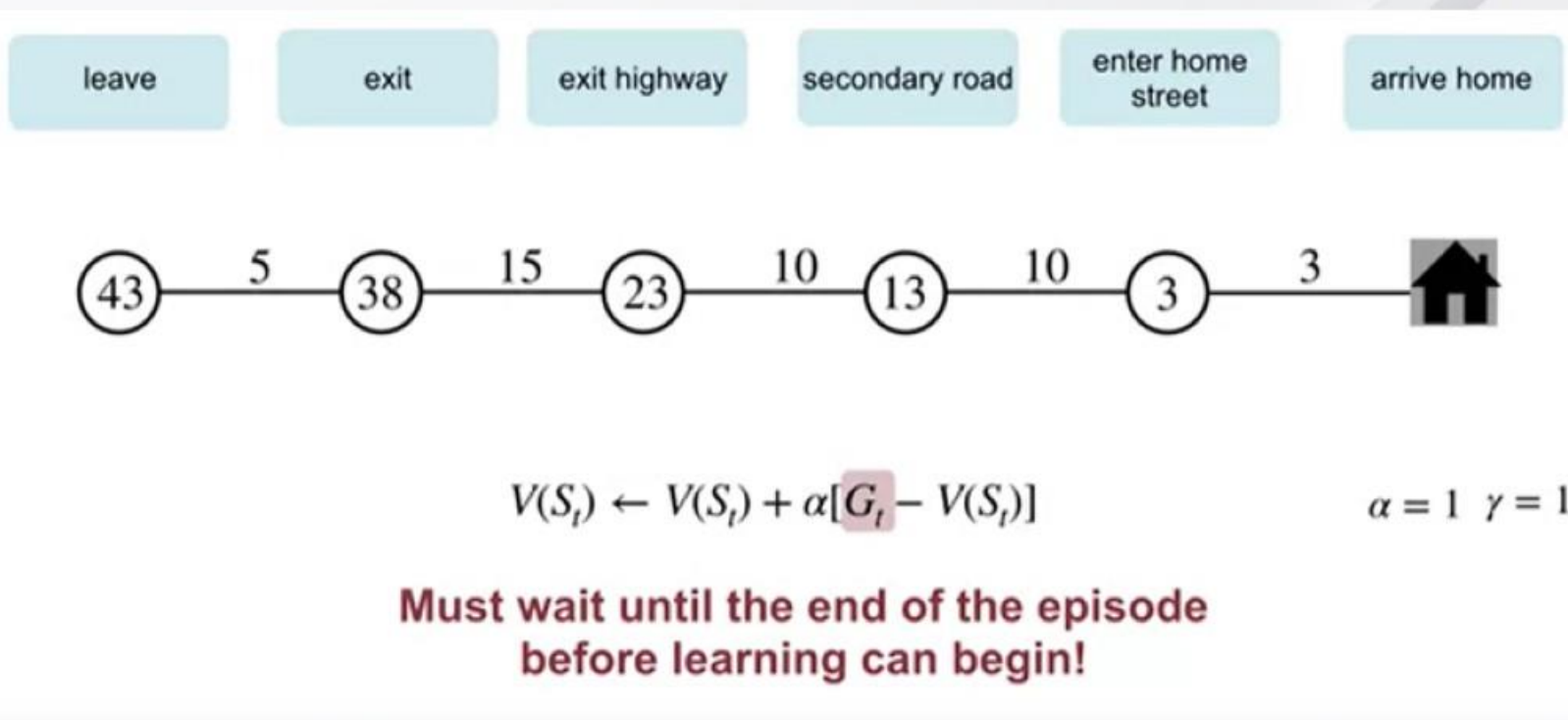
$$V(S_t) \leftarrow V(S_t) + \alpha[G_t - V(S_t)]$$

$$\alpha = 1 \quad \gamma = 1$$

$$V(\text{exit}) \leftarrow V(\text{exit}) + \alpha[G_1 - V(\text{exit})]$$

35 38 35

Example: Driving Home (Using MC)



Example: Driving Home (Using TD)

leave

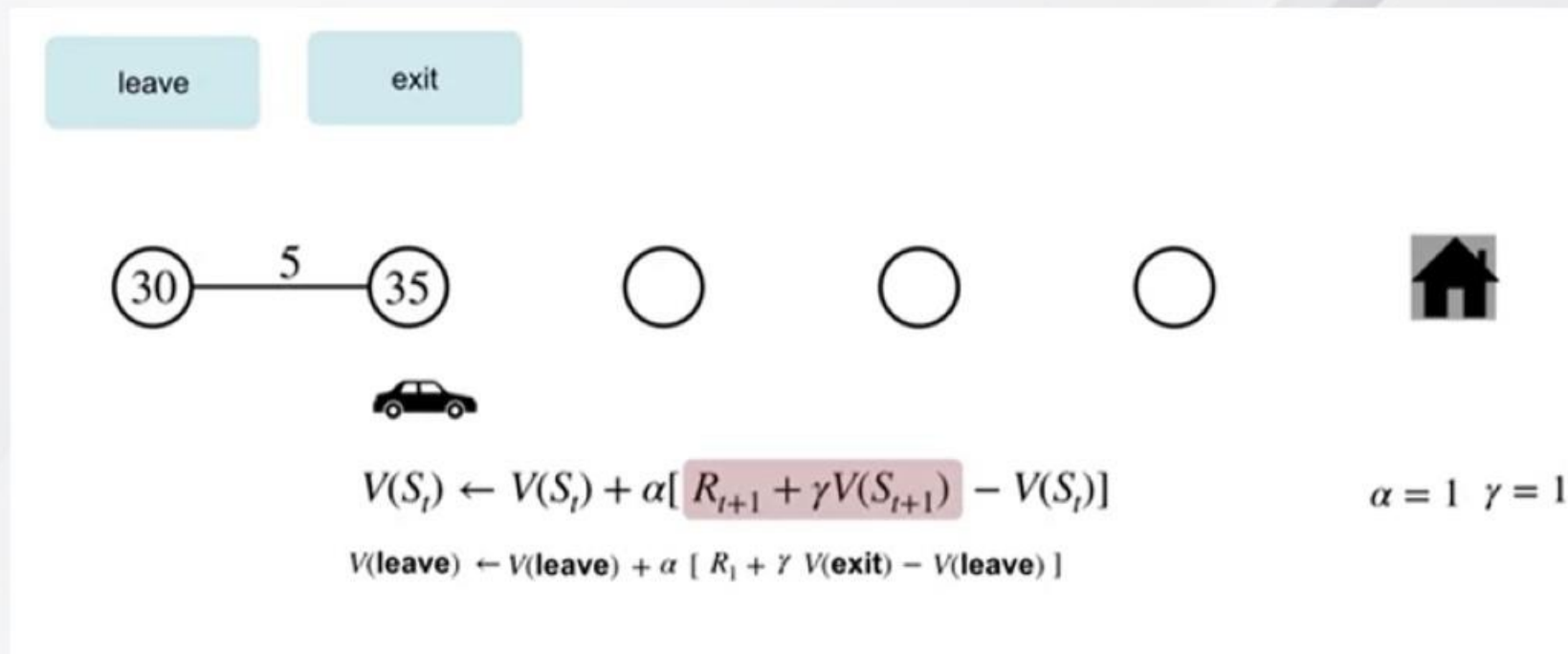
30



$$V(S_t) \leftarrow V(S_t) + \alpha [R_{t+1} + \gamma V(S_{t+1}) - V(S_t)]$$

$$\alpha = 1 \quad \gamma = 1$$

Example: Driving Home (Using TD)



Example: Driving Home (Using TD)

leave

exit



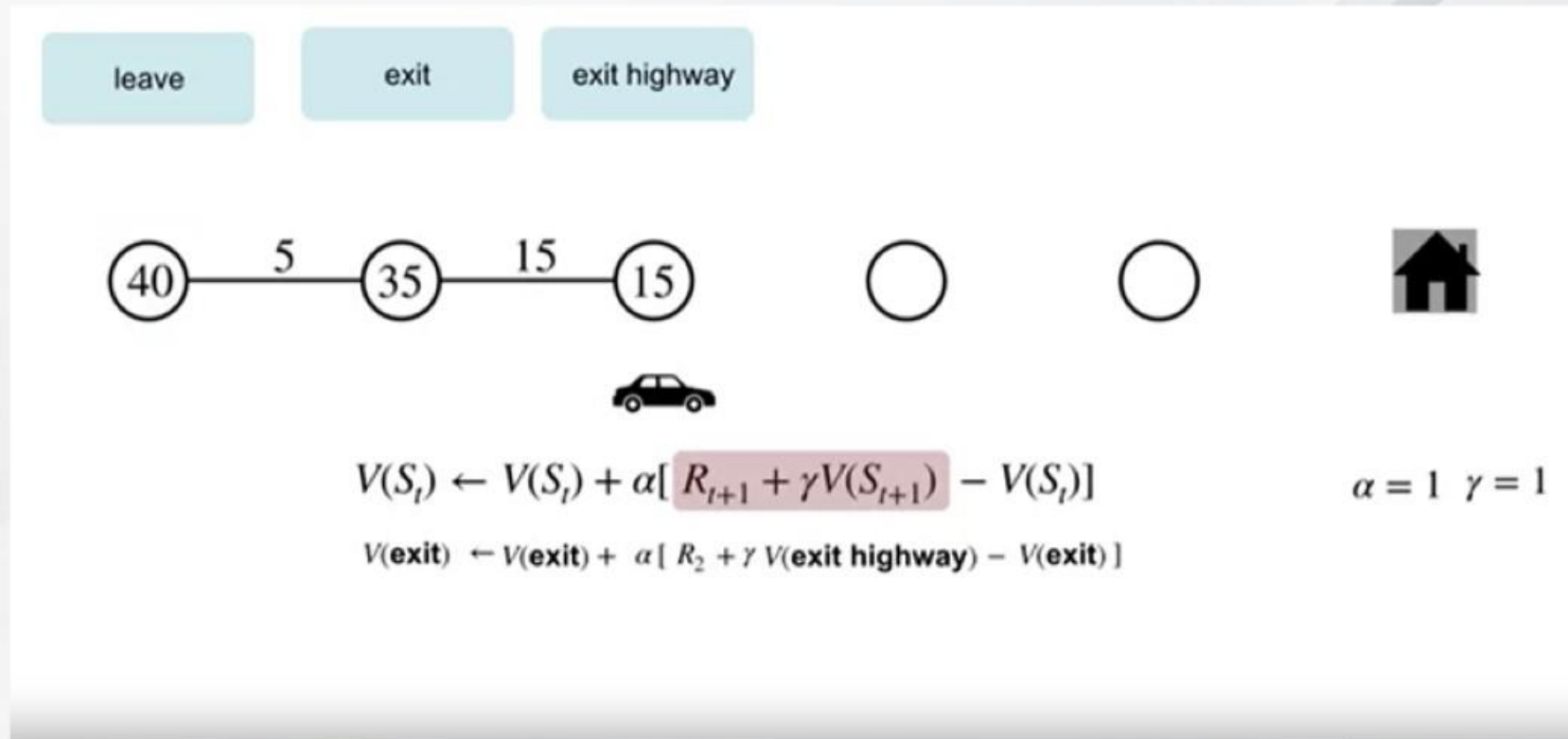
$$V(S_t) \leftarrow V(S_t) + \alpha [R_{t+1} + \gamma V(S_{t+1}) - V(S_t)]$$

$$\alpha = 1 \quad \gamma = 1$$

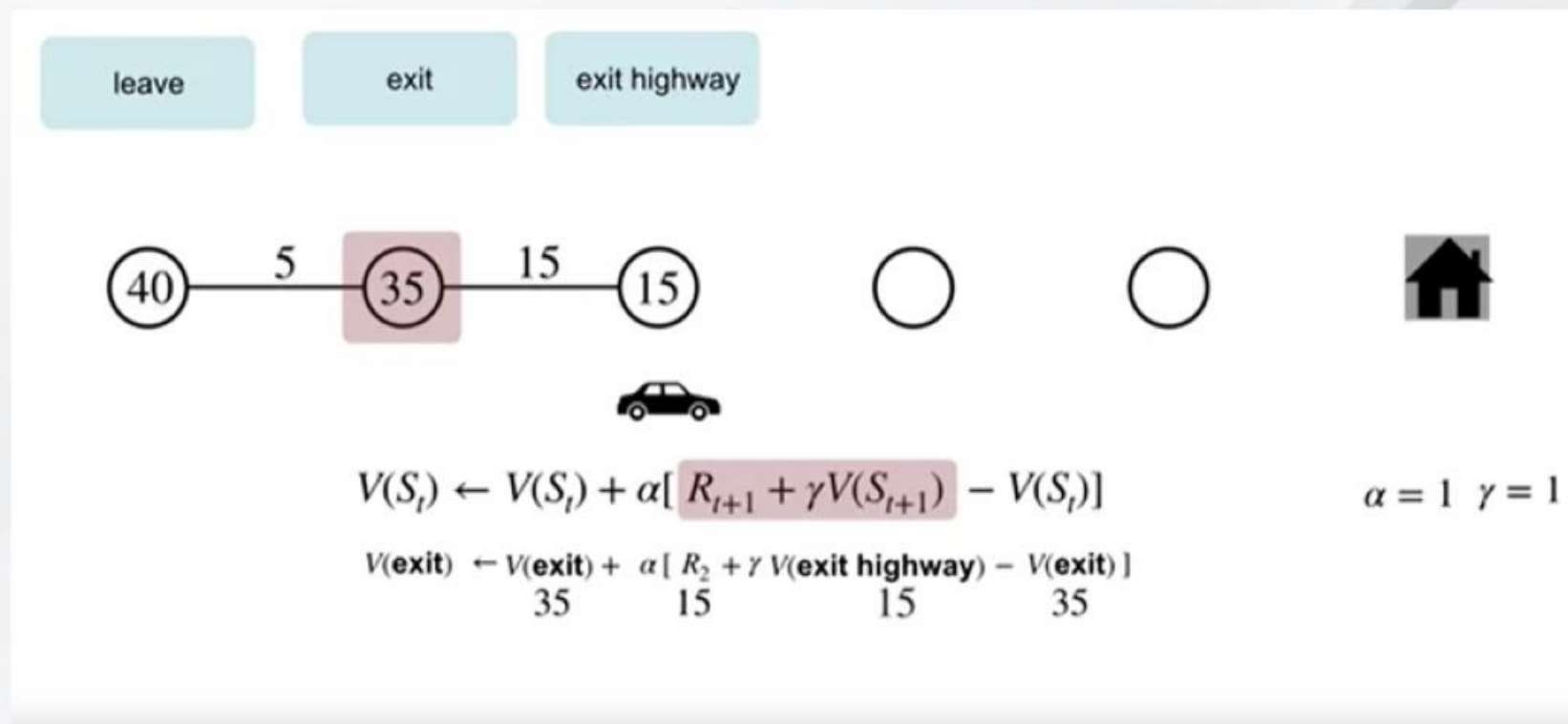
$$V(\text{leave}) \leftarrow V(\text{leave}) + \alpha [R_1 + \gamma V(\text{exit}) - V(\text{leave})]$$

30 5 35 30

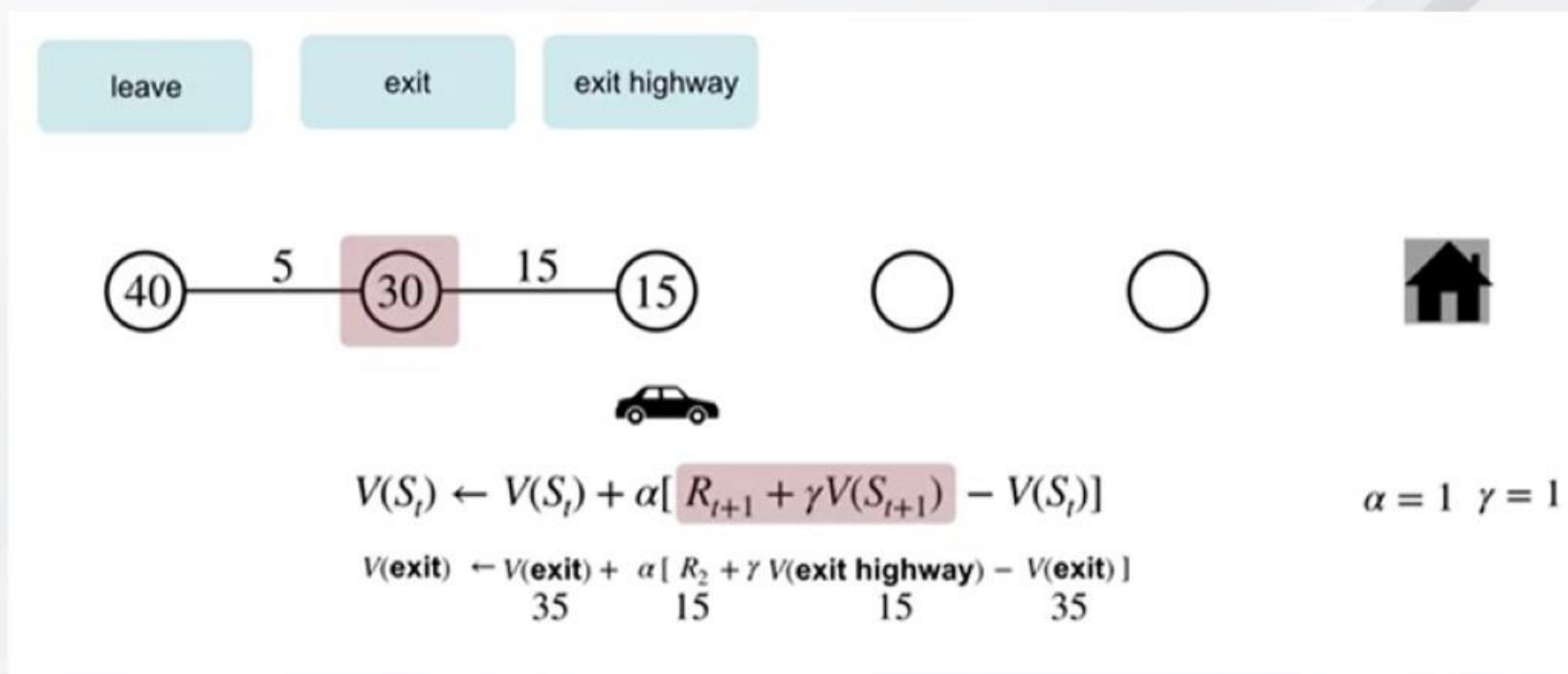
Example: Driving Home (Using TD)



Example: Driving Home (Using TD)



Example: Driving Home (Using TD)



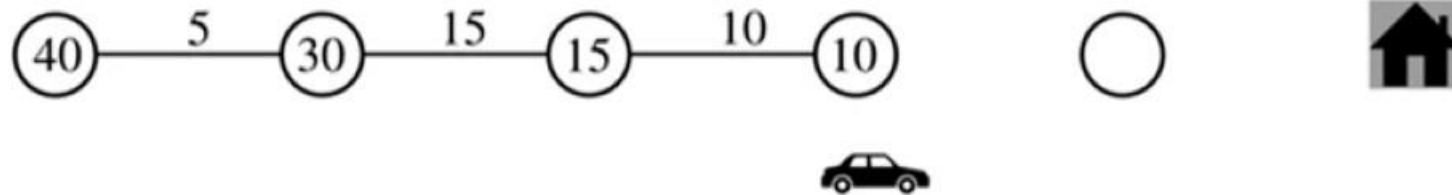
Example: Driving Home (Using TD)

leave

exit

exit highway

secondary road

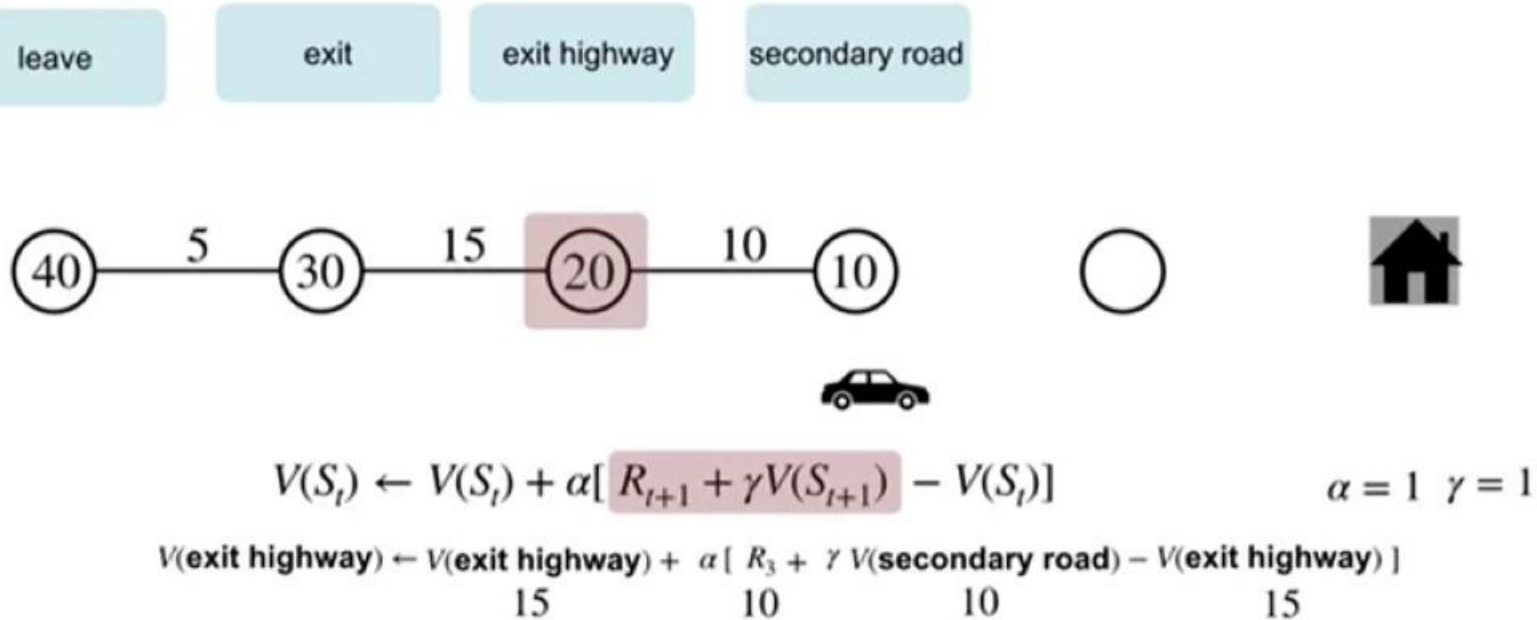


$$V(S_t) \leftarrow V(S_t) + \alpha [R_{t+1} + \gamma V(S_{t+1}) - V(S_t)]$$

$$\alpha = 1 \quad \gamma = 1$$

$$V(\text{exit highway}) \leftarrow V(\text{exit highway}) + \alpha [R_3 + \gamma V(\text{secondary road}) - V(\text{exit highway})]$$

Example: Driving Home (Using TD)



Example: Driving Home (Using TD)

leave

exit

exit highway

secondary road

enter home
street



$$V(S_t) \leftarrow V(S_t) + \alpha [R_{t+1} + \gamma V(S_{t+1}) - V(S_t)]$$

$$\alpha = 1 \quad \gamma = 1$$

$$V(\text{secondary road}) \leftarrow V(\text{secondary road}) + \alpha [R_4 + \gamma V(\text{enter home st.}) - V(\text{secondary road})]$$

Example: Driving Home (Using TD)

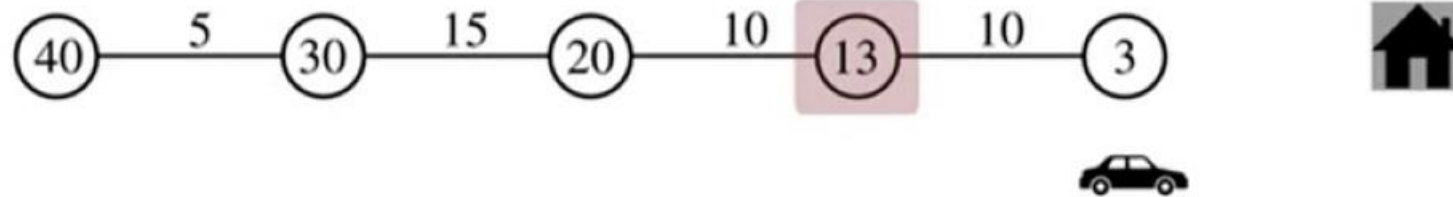
leave

exit

exit highway

secondary road

enter home
street

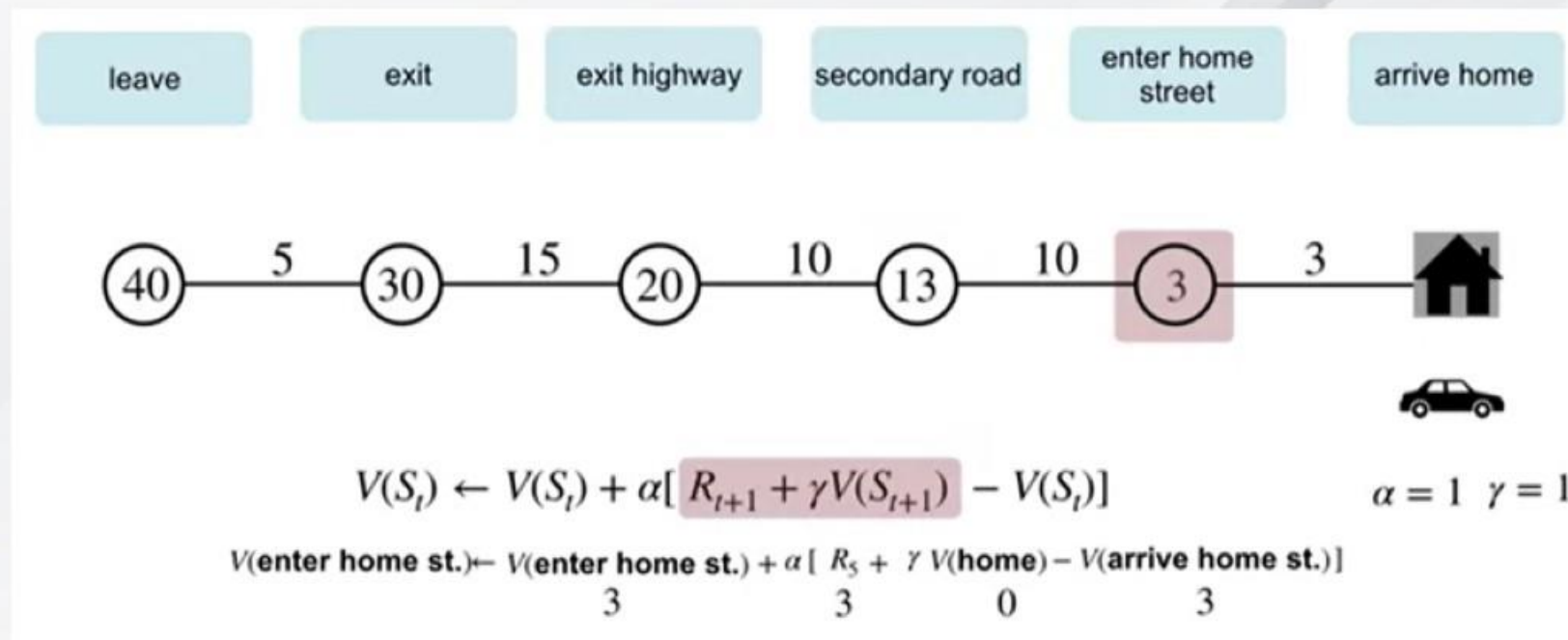


$$V(S_t) \leftarrow V(S_t) + \alpha [R_{t+1} + \gamma V(S_{t+1}) - V(S_t)]$$

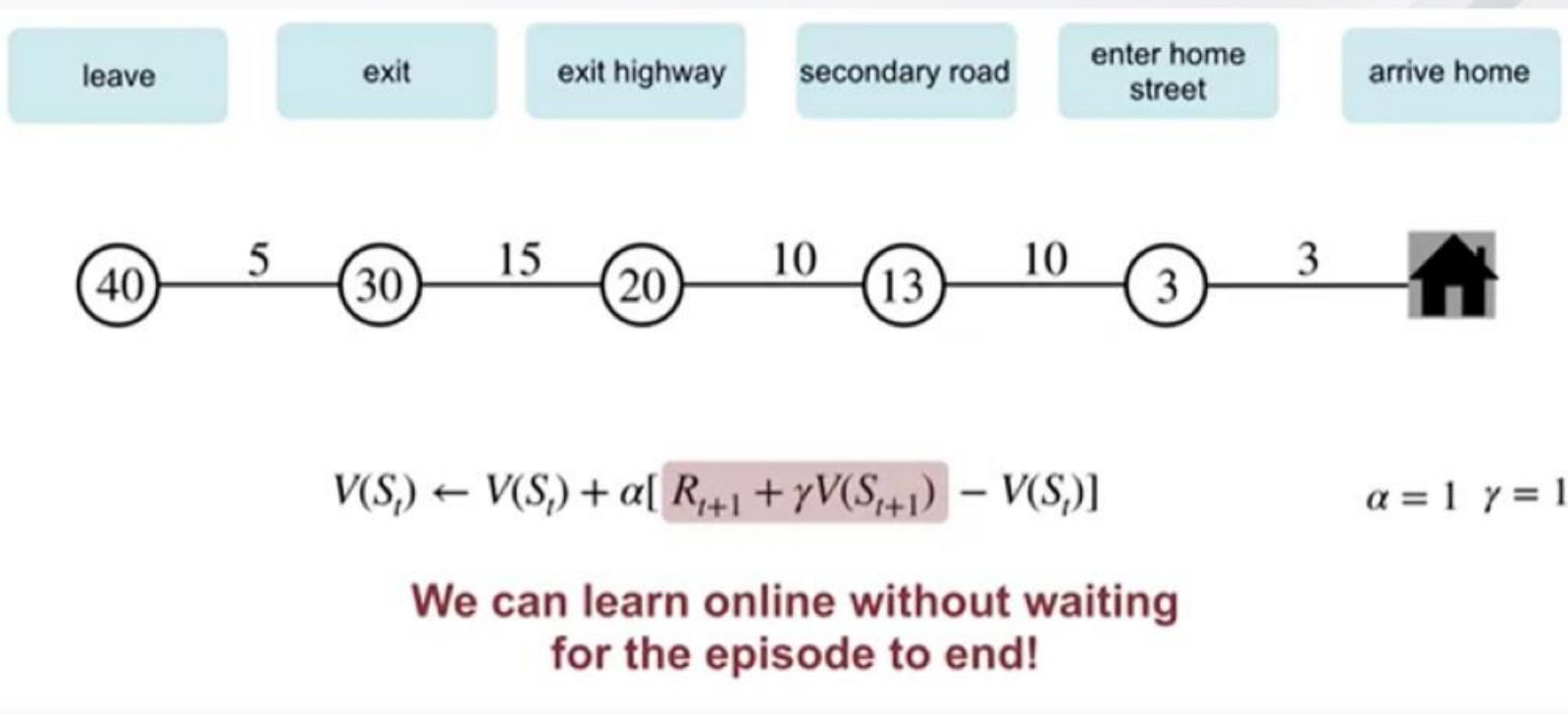
$$\alpha = 1 \quad \gamma = 1$$

$$V(\text{secondary road}) \leftarrow V(\text{secondary road}) + \alpha \left[\underset{10}{R_4} + \underset{10}{\gamma} \underset{3}{V(\text{enter home st.})} - \underset{10}{V(\text{secondary road})} \right]$$

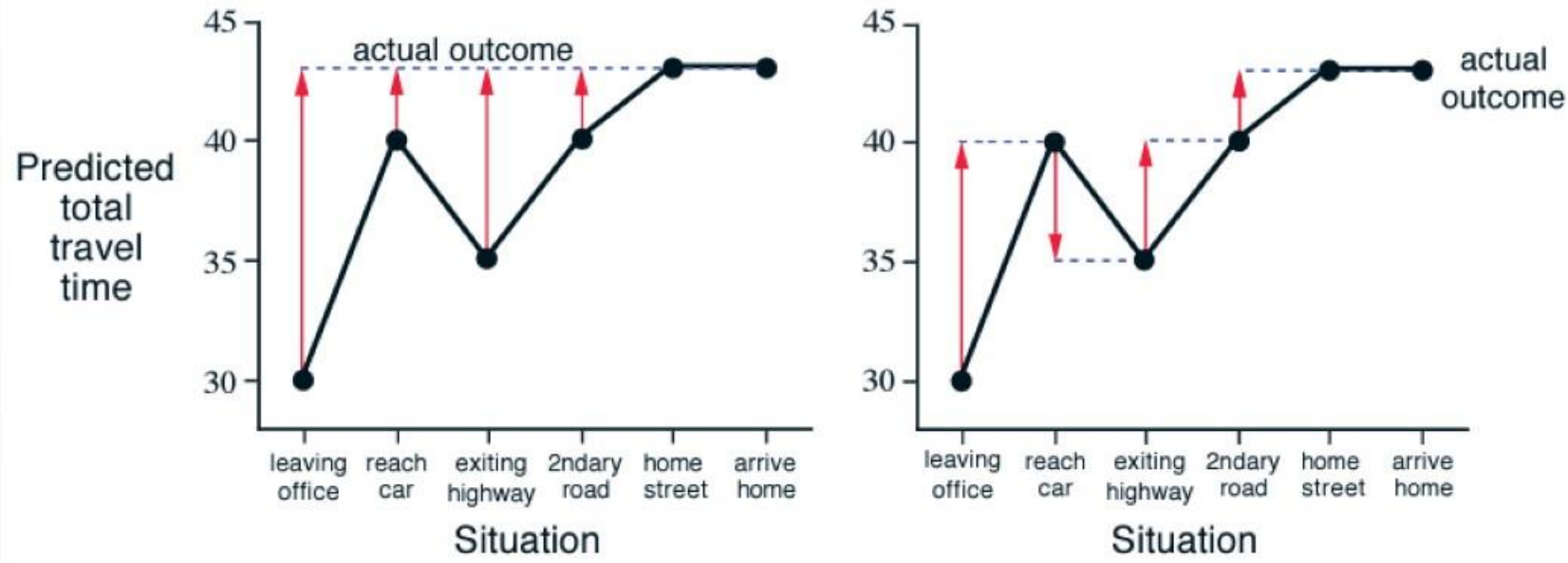
Example: Driving Home (Using TD)



Example: Driving Home (Using TD)



Unlike MC



Changes recommended in the driving home example by Monte Carlo methods (left) and TD methods (right).

The advantages of TD

- Do not require a model of the environment (Unlike Dynamic Programming)
- Updates estimates based on other estimates on every step (Unlike Monte Carlo)
- Converges faster than Monte Carlo methods

Comparing TD and MC

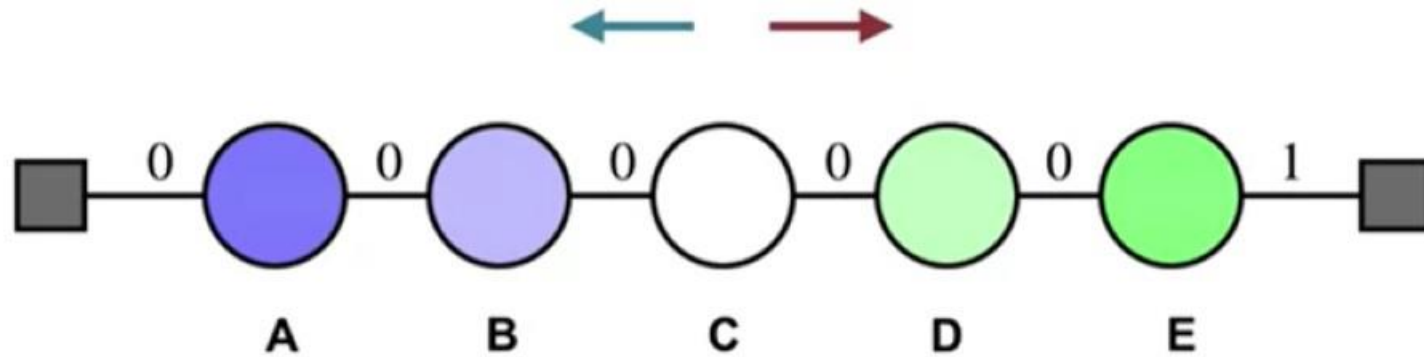
In this example we empirically compare the prediction abilities of TD(0) and constant- α MC when applied to the following Markov reward process:



A *Markov reward process*, or MRP, is a Markov decision process without actions. We will often use MRPs when focusing on the prediction problem, in which there is no need to distinguish the dynamics due to the environment from those due to the agent. In this MRP, all episodes start in the center state, C, then proceed either left or right by one state on each step, with equal probability. Episodes terminate either on the extreme left or the extreme right. When an episode terminates on the right, a reward of +1 occurs; all other rewards are zero. For example, a typical episode might consist of the following state-and-reward sequence: C, 0, B, 0, C, 0, D, 0, E, 1. Because this task is undiscounted, the true value of each state is the probability of terminating on the right if starting from that state. Thus, the true value of the center state is $v_{\pi}(C) = 0.5$. The true values of all the states, A through E, are $\frac{1}{6}$, $\frac{2}{6}$, $\frac{3}{6}$, $\frac{4}{6}$, and $\frac{5}{6}$.

Comparing TD and MC

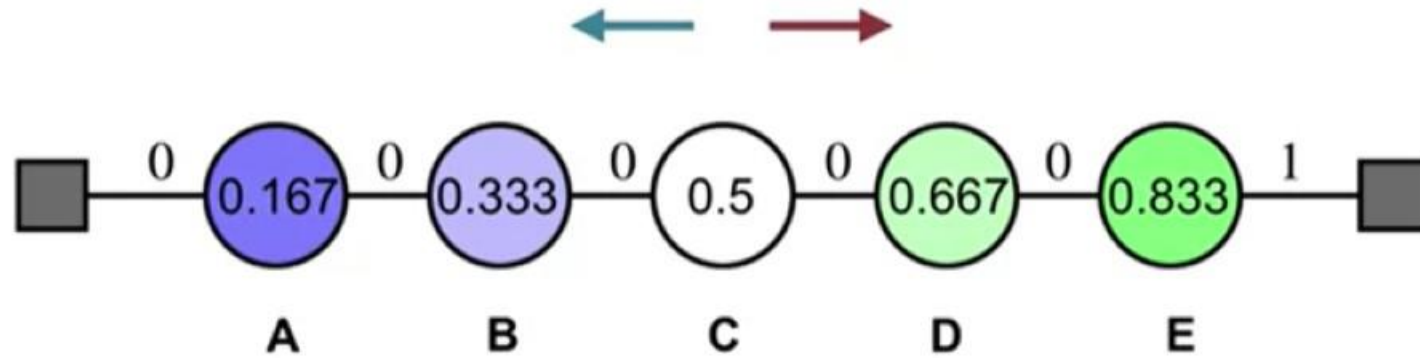
Random Walk



$$\pi(. | s) = 1/2 \quad \forall s \in \mathcal{S} \quad \gamma = 1$$

Comparing TD and MC

Random Walk

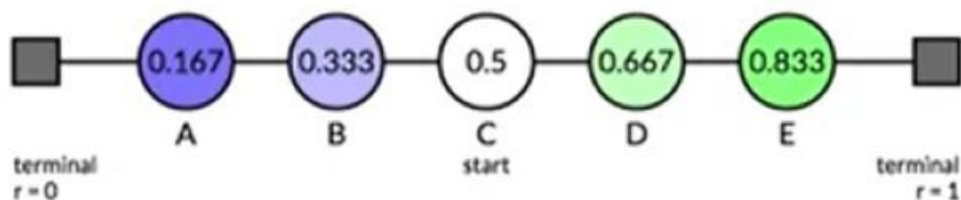


$$\pi(\cdot | s) = 1/2 \quad \forall s \in \mathcal{S} \quad \gamma = 1$$

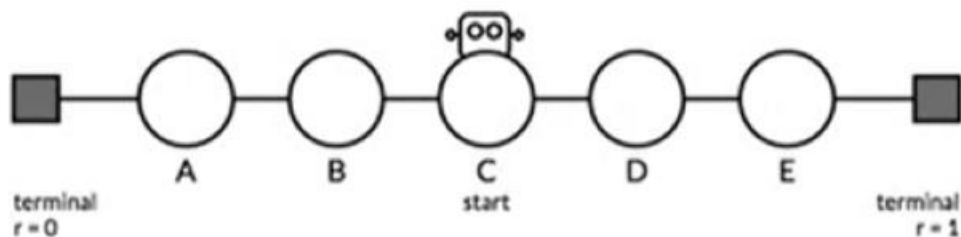
Comparing TD and MC

$\alpha=0.5$

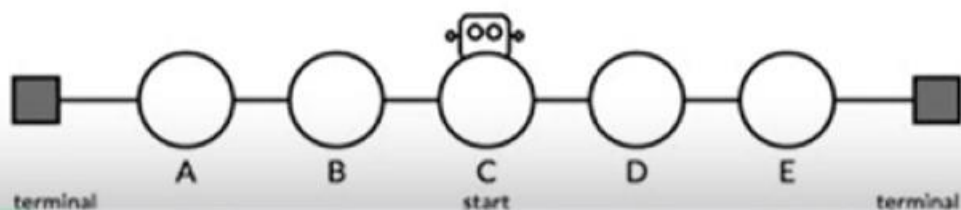
Target / Exact Values



Updates using TD Learning



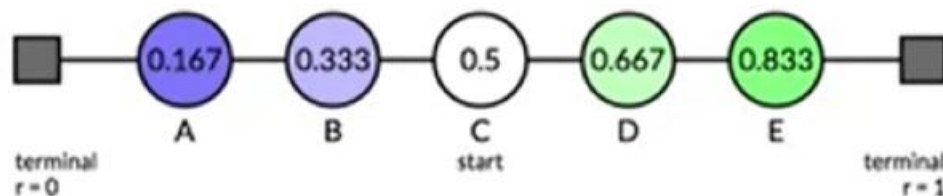
Updates using Monte Carlo



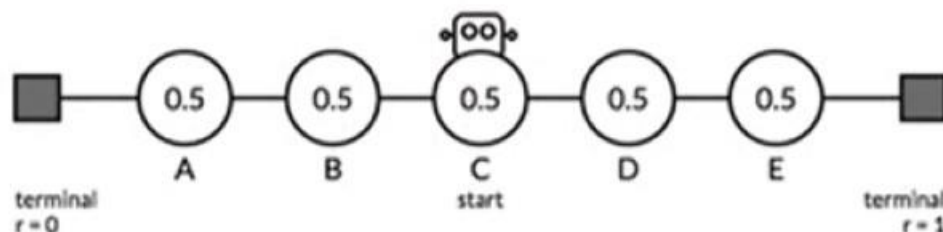
Comparing TD and MC

$\alpha=0.5$

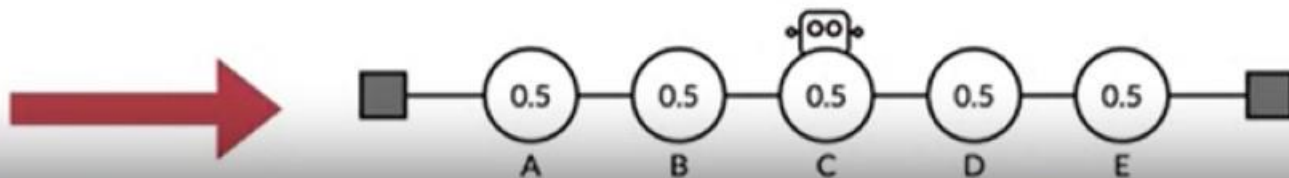
Target / Exact Values



Updates using TD Learning $V(S_t) \leftarrow V(S_t) + \alpha [R_{t+1} + \gamma V(S_{t+1}) - V(S_t)]$



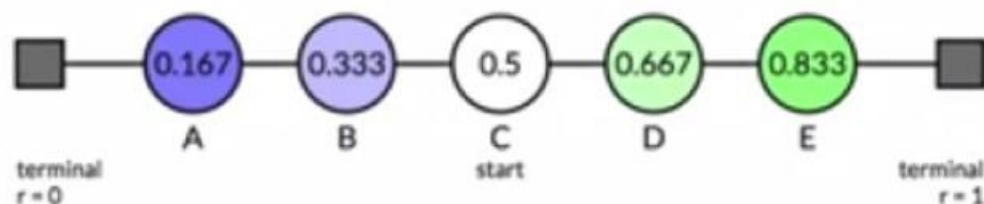
Updates using Monte Carlo



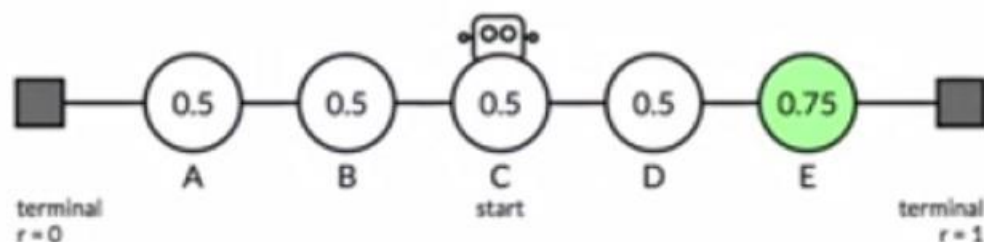
Comparing TD and MC

$\alpha=0.5$

Target / Exact Values



Updates using TD Learning $V(S_t) \leftarrow V(S_t) + \alpha [R_{t+1} + \gamma V(S_{t+1}) - V(S_t)]$



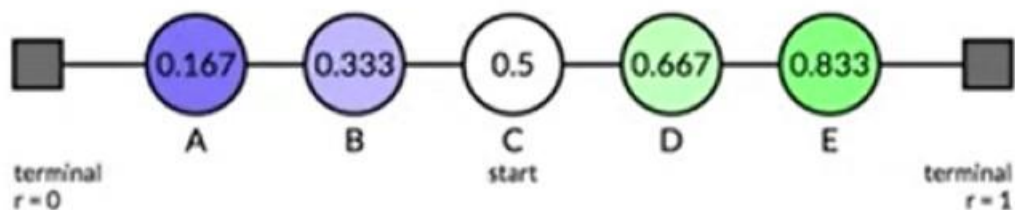
Updates using Monte Carlo $V(S_t) \leftarrow V(S_t) + \alpha [G_t - V(S_t)]$



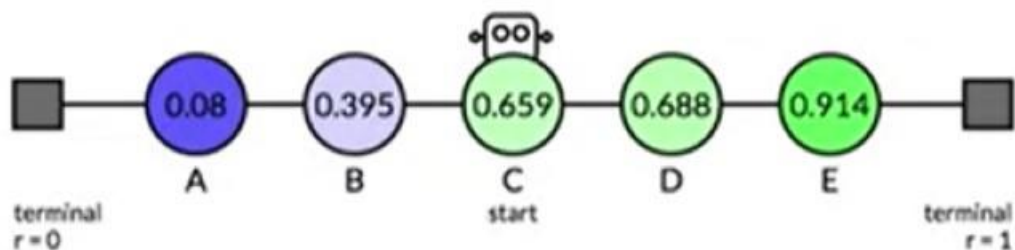
Comparing TD and MC

$\alpha=0.5$

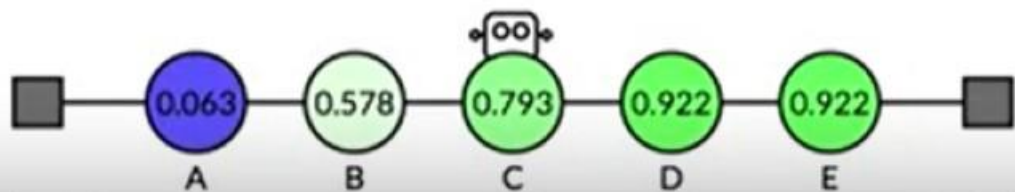
Target / Exact Values



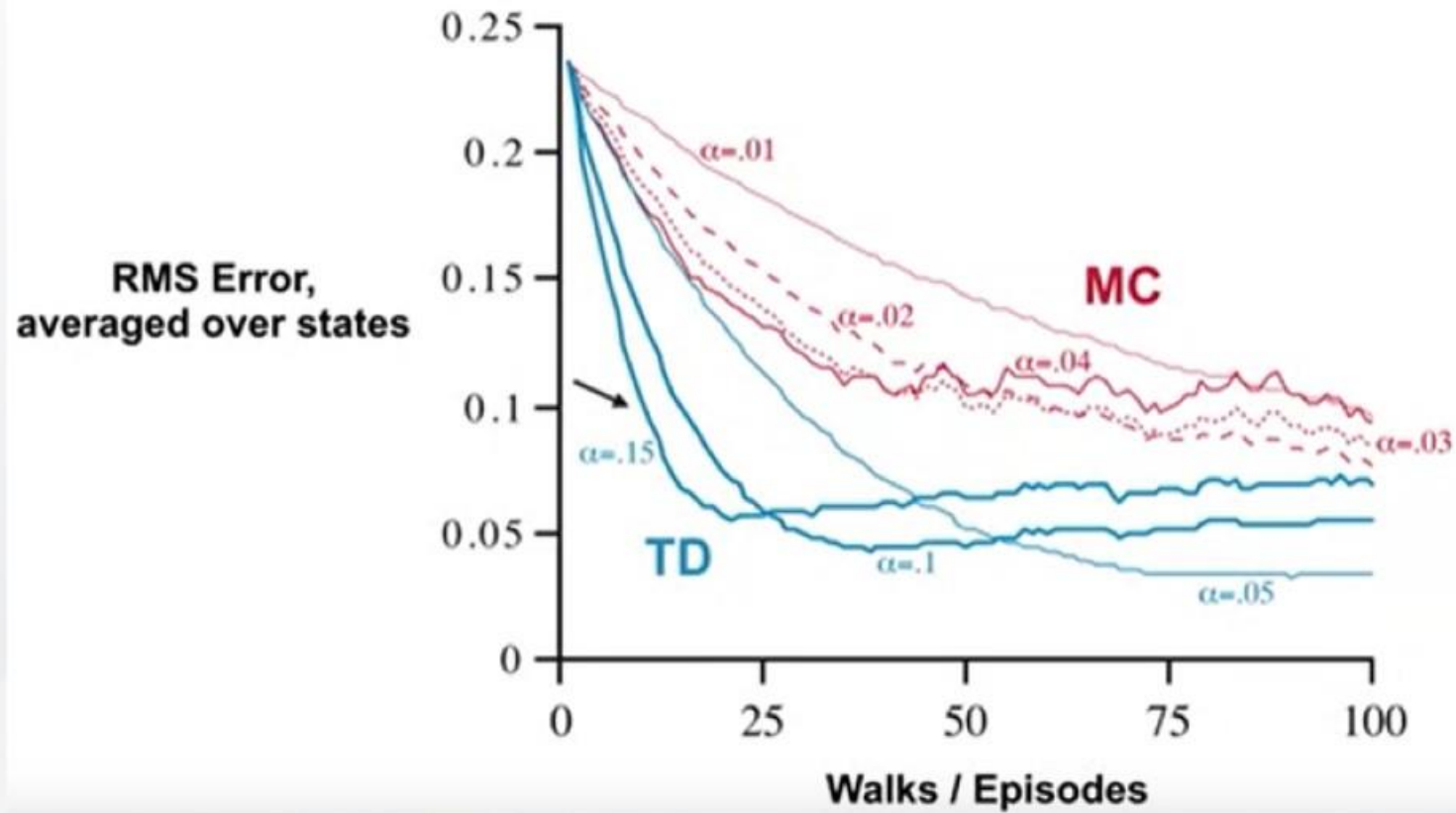
Updates using TD Learning



Updates using Monte Carlo



Comparing TD and MC



TD as an adjusted version of MC

GPI has two components:

evaluation

improvement

The adjustment is within **evaluation**.

- We may study an MRP over an MDP.
- We may reference $V(s)$ instead of $Q(s, a)$.

It addresses a disabling feature of MC:

*An episode must complete before values
can be updated.*

If episodes are long, learning may be too slow.

Example

-1	-1	-1
-1	-1	-1
-1	-10	+10

	-1	-1
-1	-1	-1
-1	-10	+10

	-1	-1
-1	-1	-1
-1	-10	+10

s	V(s)
S1	0
S2	0
S3	0
.	.
.	.
Sn	0

S1			-1	-1
	-1	-1	-1	
	-1	-10	+10	

s	V(s)
S1	0
S2	0
S3	0
.	.
.	.
Sn	0

S1	-1	→ 	-1
	-1	-1	-1
	-1	-10	+10

s	V(s)
S1	0
S2	0
S3	0
.	.
.	.
Sn	0

-1	S2 	-1
-1	-1	-1
-1	-10	+10

s	V(s)
S1	0
S2	0
S3	0
·	·
·	·
Sn	0

$$V(S_1)_{observed} = R(S_2) + V(S_2) = -1$$

$$= -1 + 0$$

$$\text{Temporal Difference Error} = V(S_1)_{\text{observed}} - V(S_1)_{\text{expected}}$$

$$= -1 - 0 = -1$$

$$V(S_1) = V(S_1) + \alpha \times \text{Temporal Difference Error}$$

$$\alpha \text{ learning rate} = 0.1 \text{ (say)}$$

$$= 0 + 0.1 \times -1 = -0.1$$

$$V(S_1) = V(S_1) + \alpha \times \text{Temporal Difference Error}$$

$$= 0 + 0.1 \times -1 = -0.1$$

s	V(s)
S1	-0.1
S2	0
S3	0
.	.
.	.
Sn	0

-1		-1
-1	-1	-1
-1	-10	+10

s	V(s)
S1	-0.1
S2	0
S3	0
.	.
.	.
Sn	0

-1	-1	-1
-1	S_7 	-1
-1	-10	+10

s	$V(s)$
S_1	-0.1
S_2	0
S_3	0
.	.
.	.
S_n	0

$$V(S_2)_{observed} = R(S_7) + V(S_7)$$

$$V(S_2)_{observed} = R(S_7) + V(S_7) = -1$$

$$= -1 + 0$$

$$\text{Temporal Difference Error} = V(S_2)_{observed} - V(S_2)_{expected}$$

$$= -1 - 0 = -1$$

$$V(S_2) = V(S_2) + \alpha \times \text{Temporal Difference Error}$$

$$= 0 + 0.1 \times -1 = -0.1$$

s	V(s)
S1	-0.1
S2	-0.1
S3	0
.	.
.	.
Sn	0

-1	-1	-1
-1	-1	-1
-1	-10	

s	V(s)
S1	-1.3
S2	0.32
S3	0.15
.	.
.	.
Sn	0.92

Solved Example

$\alpha = 0.9$ $\gamma = 1$ π - random

Initialize

d	e	f
a	b	c



0	0	0
0	0	0

$$V(s) \leftarrow V(s) + \alpha[r + \gamma V(s') - V(s)]$$

$$V(s) \leftarrow V(s) + \alpha[r + \gamma V(s') - V(s)]$$

$\alpha = 0.9$ $\gamma = 1$ π - random

Initialize

d	e	f
a	b	c



0	0	0
0	0	0

$c \rightarrow f$ $V(c) \leftarrow 0 + 0.9[100 + 0 - 0] = 90$

0	0	0
0	0	0



0	0	0
0	0	90

$$V(s) \leftarrow V(s) + \alpha[r + \gamma V(s') - V(s)]$$

$$a \rightarrow b \quad V(a) \leftarrow 0 + 0.9[0 + 0 - 0] = 0$$

0	0	0
0	0	90



0	0	0
0	0	90

$$e \rightarrow f \quad V(e) \leftarrow 0 + 0.9[100 + 0 - 0] = 90$$

0	0	0
0	0	90



0	90	0
0	0	90

$$V(s) \leftarrow V(s) + \alpha[r + \gamma V(s') - V(s)]$$

$$b \rightarrow c \quad V(b) \leftarrow 0 + 0.9[0 + 90 - 0] = 81$$

0	90	0
0	0	90



0	90	0
0	81	90

$$d \rightarrow e \quad V(d) \leftarrow 0 + 0.9[0 + 90 - 0] = 81$$

0	90	0
0	81	90



81	90	0
0	81	90

$$V(s) \leftarrow V(s) + \alpha[r + \gamma V(s') - V(s)]$$

$$a \rightarrow b \quad V(a) \leftarrow 0 + 0.9[0 + 81 - 0] \approx 73$$

81	90	0
0	81	90



81	90	0
73	81	90

$$c \rightarrow f \quad V(c) \leftarrow 90 + 0.9[100 + 0 - 90] = 99$$

81	90	0
73	81	90



81	90	0
73	81	99

$$V(s) \leftarrow V(s) + \alpha[r + \gamma V(s') - V(s)]$$

$$e \rightarrow f \quad V(e) \leftarrow 90 + 0.9[100 + 0 - 90] = 99$$

81	90	0
73	81	99



81	99	0
73	81	99

$$c \rightarrow b \quad V(c) \leftarrow 99 + 0.9[0 + 81 - 99] \approx 83$$

81	99	0
73	81	99



81	99	0
73	81	83

Summary

- Define temporal-difference learning
- Define the temporal-difference error
- Understand the TD(0) algorithm
- Understand the benefits of learning online with TD
- Identify key advantages of TD methods over Dynamic Programming and Monte Carlo methods



Thank You