# Verslag labo 2

1. **Question 1: Value Iteration**

Placeholder.

1. **Question 2: Bridge Crossing Analysis**

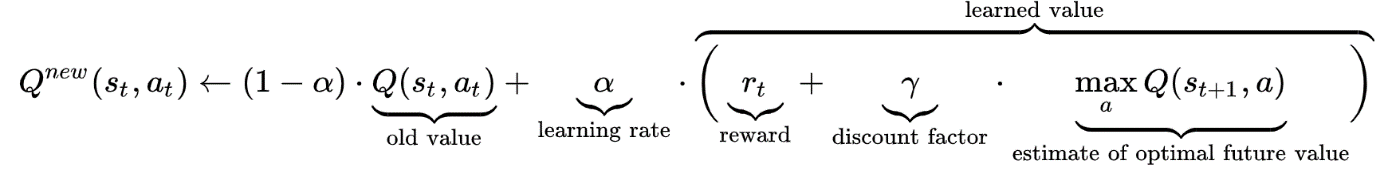
Placeholder.

1. **Question 3: Policies**

Placeholder.

1. **Question 6: Q-Learning**

Q-Learning, in tegenstelling tot Value Iteration in opdracht 1, leert wel uit ervaring. In een Q-tabel wordt, voor elke combinatie van action en state, een Q-value bijgehouden. In het begin zijn deze waarden 0. Na een aantal keer spelen, wordt de tabel stelselmatig aangevuld. De Agent kiest zijn volgende actie door het zoeken van de hoogste Q-value in de huidige state.

In “QLearningAgent” werden een aantal methodes vervolledigd. “Learning” in Q-Learning gebeurt voornamelijk in update(state, action, nextState, reward). Hier krijgt een action, state paar een nieuwe Q-value aangewezen. De Agent kan deze later gebruiken voor het maken van een beslissing. De gebruikt formule is gebaseerd op deze van Wikipedia:

De laatste term in de vorige formule werd geïmplementeerd in “computeValueFromQValues”. De Agent gaat altijd naar de volgende state met de hoogste Q-Value. Elke Q-value, na het nemen van een actie, wordt in een lijst gestoken. Na het doorlopen van alle “legale” acties, kan de maximum Q-value van de lijst teruggegeven worden aan het programma. “ComputeActionfromQValues” geeft de actie terug, bijhorend bij de maximum Q-value.

1. **Question 7: Epsilon Greedy**

Placeholder.

1. **Question 8: Bridge Crossing Revisited**

Placeholder.

1. **Question 9: Q-Learning and Pacman**

Placeholder.

1. **Question 10: Approximate Q-Learning**

Placeholder.