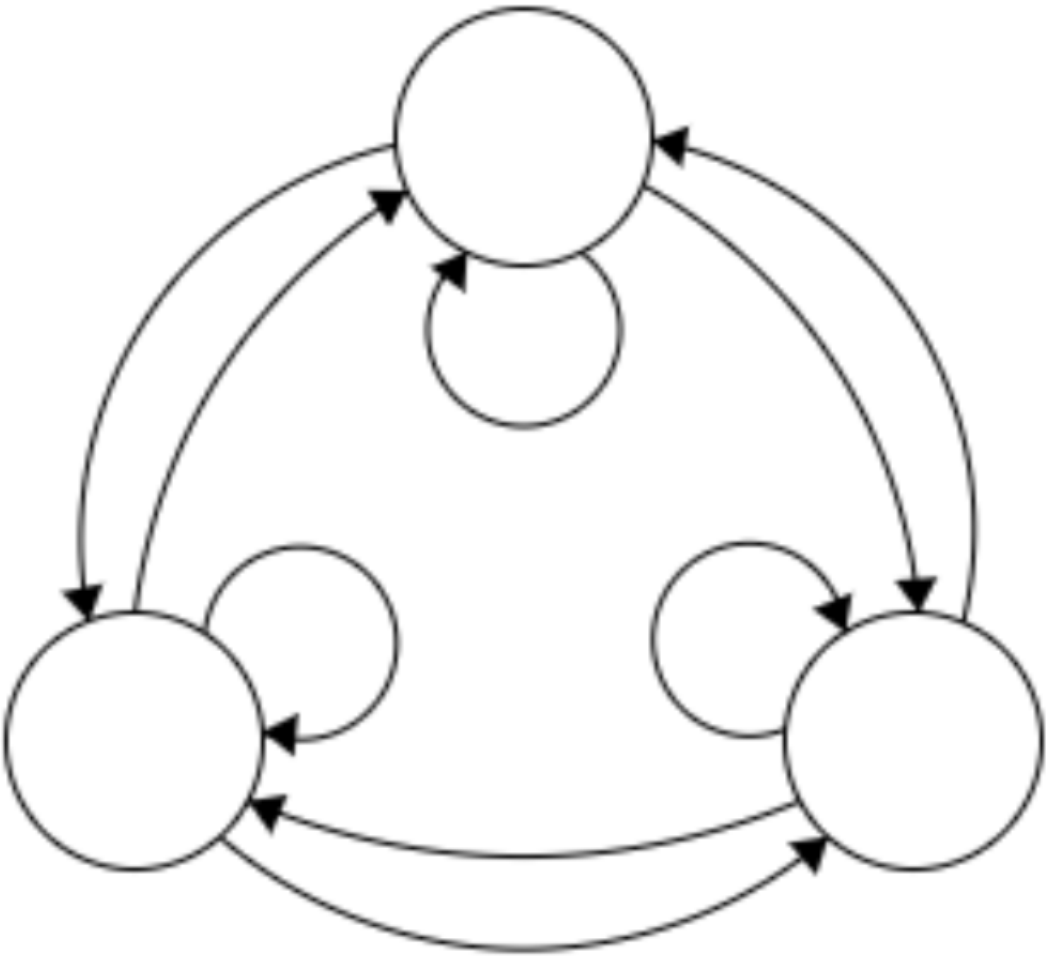
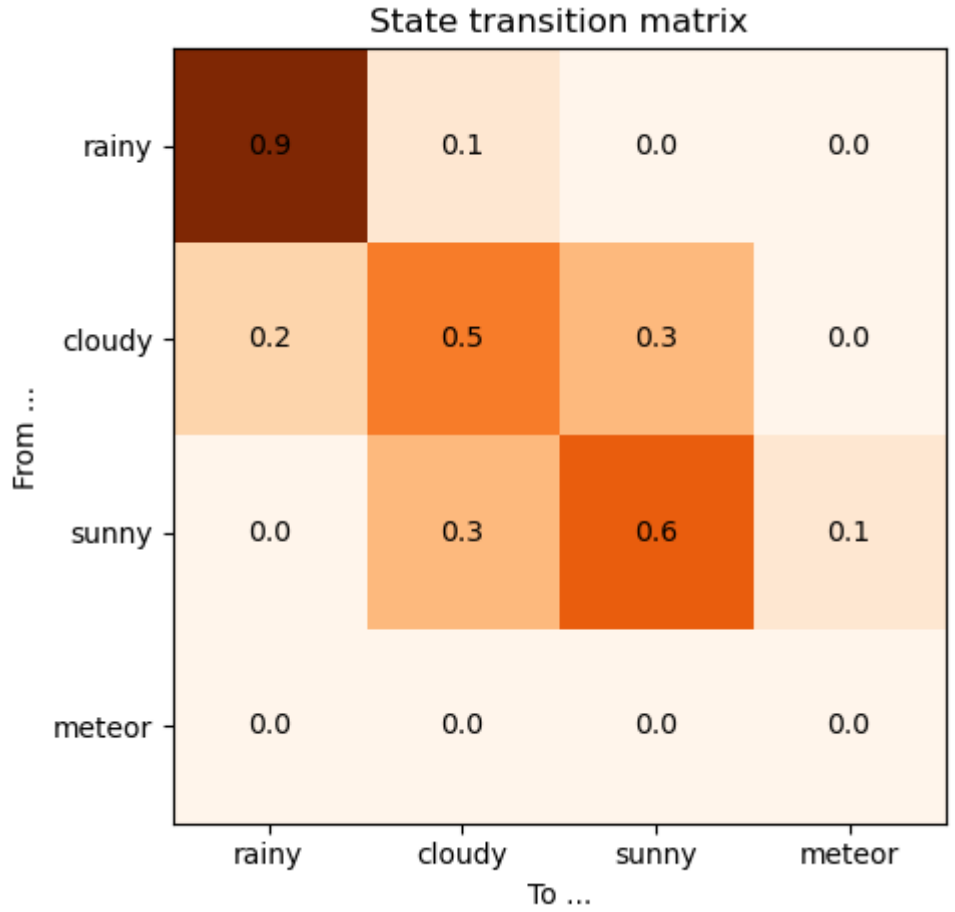
Model-based prediction & control



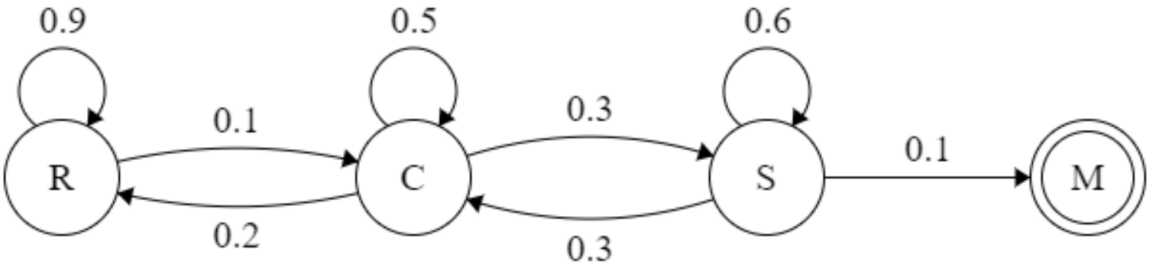
John Williams - 1790472

A. Markov Chain

Opdracht: *Teken een Markov Chain van 4 states aan de hand van deze state transition probability matrix <S, P>; S = states, P = probabilities. Er is geen specifieke beginstate. Er is wel een specifieke eindstate. Zorg ervoor dat visueel duidelijk is welke dit is.*



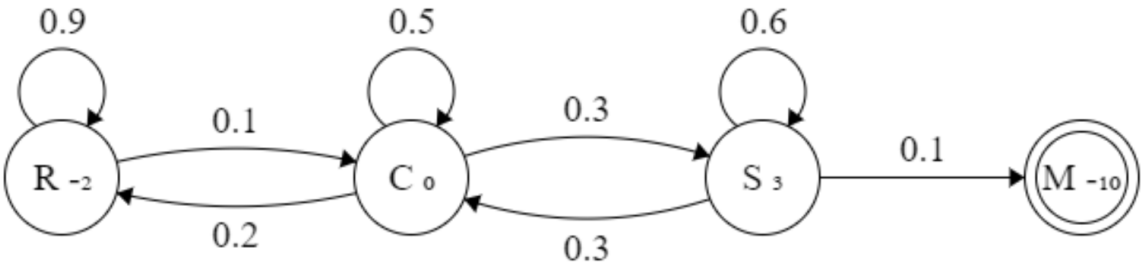
De state transition probability matrix in kwestie



De Markov Chain, vanuit de bovenstaande matrix. De state symbolen volgen uit de eerste letter van iedere state in de state transition probability matrix.

B. Markov Reward Process

Opdracht: *Maak van de Markov Process een Markov Reward Process. De states (van links naar rechts) hebben respectievelijk rewards -2, 0, 3 en -10. Zie het zo: naar de state 'rain' toe gaan geeft altijd een reward van -2, waar je ook vandaan komt. Het maakt voor de berekeningen niet uit of je zegt dat in een bepaalde state zijn of naar een bepaalde state toegaan de reward geeft. Dus doe wat voor jou intuïtief is.*



De reward van iedere state is nu naast het symbool van de state geschreven.

C. Sampling

Opdracht: *Pak twee mogelijke samples van je MRP en leg uit wat de return was voor elke sample.*

De formule voor het berekenen van de return: $G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$

De gekozen samples en de bijbehorende return zijn als volgt:
CRCSSM:

$$G_t = (1^1 \cdot -2) + (1^2 \cdot 0) + (1^3 \cdot 3) + (1^4 \cdot 3) + (1^5 \cdot -10)$$
$$= (-2) + 0 + 3 + 3 + (-10)$$
$$= -6$$

RCCSM:

$$G_t = (1^1 \cdot 0) + (1^2 \cdot 0) + (1^3 \cdot 3) + (1^4 \cdot -10)$$
$$= 0 + 0 + 3 + (-10)$$
$$= -7$$

Bij de bovenstaande berekeningen zijn de eerste regels de summatie volledig uitgeschreven, waarbij ieder groep haakjes 1 iteratie is van de summatie.

D. Value Function

Opdracht: *Bepaal nu voor alle states wat de value is na 2 iteraties. De value voor alle states worden geinitialiseerd op 0. Gebruik hiervoor de Bellman expectation equation met γ = 1.*

De Bellman expectation equation die gebruikt wordt is als volgt:

$$V_n = P \cdot (R + \gamma V_{n-1})$$

Waarbij:

- V_n = een vector met alle values van de states na n iteraties
- R = een vector met alle rewards van de states
- γ = een scalar, de discount (in dit geval gelijk aan 1)
- P = een matrix, de state transition probability matrix

$$V_0 = \begin{bmatrix} 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \end{bmatrix}$$

$$V_1 = \begin{bmatrix} 0.9 & 0.1 & 0.0 & 0.0 \\ 0.2 & 0.5 & 0.3 & 0.0 \\ 0.0 & 0.3 & 0.6 & 0.1 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix} \cdot \left(\begin{bmatrix} -2 \\ 0 \\ 3 \\ -10 \end{bmatrix} + \gamma V_0 \right) = \begin{bmatrix} 0.9 & 0.1 & 0.0 & 0.0 \\ 0.2 & 0.5 & 0.3 & 0.0 \\ 0.0 & 0.3 & 0.6 & 0.1 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix} \cdot \begin{bmatrix} -2 \\ 0 \\ 3 \\ -10 \end{bmatrix}$$

$$V_1 = \begin{bmatrix} -1.8 \\ 0.5 \\ 0.8 \\ 0.0 \end{bmatrix}$$

$$V_2 = \begin{bmatrix} 0.9 & 0.1 & 0.0 & 0.0 \\ 0.2 & 0.5 & 0.3 & 0.0 \\ 0.0 & 0.3 & 0.6 & 0.1 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix} \cdot \left(\begin{bmatrix} -2 \\ 0 \\ 3 \\ -10 \end{bmatrix} + \gamma V_1 \right) = \begin{bmatrix} 0.9 & 0.1 & 0.0 & 0.0 \\ 0.2 & 0.5 & 0.3 & 0.0 \\ 0.0 & 0.3 & 0.6 & 0.1 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix} \cdot \begin{bmatrix} -3.8 \\ 0.5 \\ 3.8 \\ -10 \end{bmatrix}$$

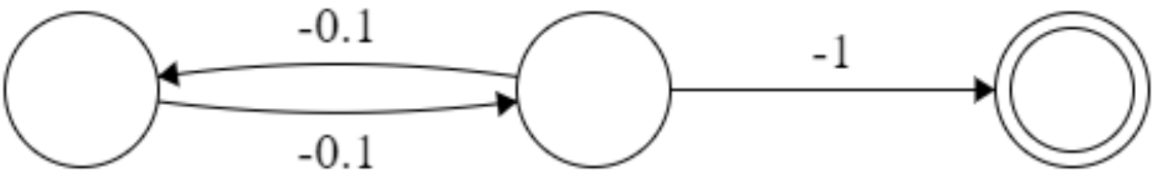
$$V_2 = \begin{bmatrix} -3.37 \\ 0.63 \\ 1.43 \\ 0.0 \end{bmatrix}$$

Opdracht: *In werkelijkheid zien we vrijwel nooit een discount factor van γ = 1. Een discount factor van γ = 0.9 is wel veelvoorkomend. Noem twee mogelijke problemen als γ = 1.*

1. In het geval van oneindige processen met een discount factor van 1 betekent het dat de return oneindig optelt en dus oneindig groot wordt.
2. Als de discount altijd 1 blijft, dan heeft de reward van de huidige tijdsstip even veel impact als de reward van een tijdsstip ver in de toekomst. Dit is vaak ongewenst omdat in non-deterministische omgevingen de toekomst niet te voorspellen is, en dus niet evenveel waard is.

E. Value Iteration

Opdracht: *Bepaal de utility van elke state in de MDP. Je moet bepalen wanneer je stopt met itereren -- beargumenteer je keuze.*



De MDP in kwesie

De utility matrix begint als volgt:

$$U_0 = \begin{matrix} & A & B & C \\ L & \end{matrix} \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{bmatrix}$$
$$R$$

De rode cellen representeren acties die niet op die state uitgevoerd kunnen worden.

Voor iedere non-rode state-action cell wordt de utility berekend door de reward van de action nemen op te tellen met de maximum waarde gevonden in de kolom van de bestemming state (rode cellen worden niet meegenomen in het kiezen van de max, behalve als het niet anders kan)

$$U_1 = \begin{bmatrix} 0.0 & -0.1 & 0.0 \\ -0.1 & -1.0 & 0.0 \end{bmatrix}$$

$$U_2 = \begin{bmatrix} 0.0 & -0.2 & 0.0 \\ -0.2 & -1.0 & 0.0 \end{bmatrix}$$

$$\vdots$$

$$U_{10} = \begin{bmatrix} 0.0 & -1.0 & 0.0 \\ -1.0 & -1.0 & 0.0 \end{bmatrix}$$

$$U_{11} = \begin{bmatrix} 0.0 & -1.1 & 0.0 \\ -1.1 & -1.0 & 0.0 \end{bmatrix}$$

$$U_{12} = \begin{bmatrix} 0.0 & -1.2 & 0.0 \\ -1.1 & -1.0 & 0.0 \end{bmatrix}$$

Bij nog meer iteraties veranderd er niets meer in de matrix, dus dit is de utility matrix van alle states.