

Exercise Sheet 1

General remarks:

- **Due date:** October 21st 12:30 (before the exercise class).
- Please submit your solutions via MOODLE. Remember to provide your matriculation number. It is necessary to hand in your solutions in groups of **three**. You may use the MOODLE forum to form groups.
- Solutions must be written in English.
- While we will publish sketches of exercise solutions, we do *not* guarantee that these sketches contain all details that are necessary to properly solve an exercise. Hence, it is recommended to attend the exercise classes.
- If you have any questions regarding the lecture or the exercise, please use the forum in MOODLE.
- Provide solutions to programming exercises in a separate plain text file. Please name the files according to the exercise they belong to (e.g. 'sheet01_ex1a.wppl'). All your programs are required to run without errors in the online interpreter available at <http://webppl.org>.

Exercise 1 (Monty Hall problem)

25P

The *Monty Hall problem*—probably one of the most famous brain teasers in history—goes as follows¹:

Suppose you are on a game show, and you are given the choice of three doors. Behind one door there is a car; behind the other two, goats. You pick a door, say No. 1, and the host, who knows what is behind the doors, opens another door, say No. 3, which has a goat. He then asks “Do you want to pick door No. 2 instead of No. 1?” Is it to your advantage to switch your choice?

While many of you might already know the answer to the problem, your task in this exercise is to solve it with the help of a WEBPPL program. More specifically, you are asked to provide WEBPPL code that models both

- the situation where the player does *not* switch his or her choice, and
- the situation where the player *does* switch.

Solve the Monty Hall problem by comparing the outcome of your models for the following host strategies! Also state the probabilities of winning the car. **Note:** *You are allowed to use the full feature set of WEBPPL*².

- (a) [15P] Monty chooses a door uniformly at random, but neither the door chosen by the player nor the one with the car behind it.
- (b) [5P] Monty always chooses the door with the lowest number which is neither the player's choice nor the door with the car behind it.
- (c) [5P] Monty chooses a door uniformly at random, but not the player's choice.

¹https://en.wikipedia.org/wiki/Monty_Hall_problem

²<https://webppl.readthedocs.io/en/master/gettingstarted.html>

Exercise 2 (Gambler's ruin problem)**25P**

Consider a coin-flipping game with two players where each player has a 50% chance of winning with each flip of the coin. Assume that player 1 initially has n_1 pennies and player 2 has n_2 pennies. After each flip of the coin the loser transfers one penny to the winner. The game ends when one player has all the pennies.

Write a WEBPPL program that models the game and use it to answer the following questions.

- (a) [10P] If the second player starts with 10 pennies, what is the minimum number of pennies the first player should bring so that she wins with probability at least 70%?
- (b) [15P] For $n_1 = 5$ and $n_2 = 5$, what is the expected number of rounds until the game ends?

Exercise 3 (Dungeons and Dragons)**25P**

Assume you are playing a Dungeons and Dragons³ game. Suppose that you have earned yourself a bonus that allows you to throw n dice with f faces (instead of only one die) and take the maximum value. For example, if you throw $n = 3$ dice with $f = 6$ faces yielding the outcomes $(1, 4, 5)$ then this particular throw counts as a 5.

- (a) [15P] Write a WEBPPL program that computes the expected value of a bonus throw, given the global variables n and f .
- (b) [3P] Let $f = 20$. For each $n \in \{2, 3, 4\}$ compute the ratio of the expected value against the number of faces. Can you spot an approximate pattern?
- (c) [7P] Lets investigate the case where $n = 3$ a bit further. Compute (by hand, not in WEBPPL) the ratio of the expected value against the number of faces f when $f \rightarrow \infty$. Compare the distance between the estimated ratio and the ratio computed in the limit.

Hint: The number of possibilities where the maximum is x is given by $3x(x-1)+1$.

Exercise 4 (1-dimensional particle dynamics)**25P**

In this exercise we study the dynamics of a uniformly accelerated particle moving along one dimension. We assume that the initial position $x(t_0)$ of the particle has a Gaussian prior distribution with mean $\mu = 0$ and variance $\sigma = 1$. For the initial velocity $v(t_0)$ and acceleration a you can assume uniform priors in the interval $[1, 10]$ and $[1, 2]$, respectively:

$$\begin{aligned}x(t_0) &\sim \mathcal{N}(0, 1) \\v(t_0) &\sim \text{Unif}(1, 10) \\a &\sim \text{Unif}(1, 2) .\end{aligned}$$

Note that the acceleration a is constant (it does not depend on t). To approximate the particle's dynamics we use the equations

$$\begin{aligned}x(t+h) &= x(t) + hv(t) \\v(t+h) &= v(t) + ha\end{aligned}$$

for some step size $h > 0$. Now let $t_0 = 0$, $t_1 = 10$ and $h = 1$. Write a WEBPPL program that models the situation and use it for the following tasks:

³https://de.wikipedia.org/wiki/Dungeons_%26_Dragons

- (a) [10P] Give an estimate of the expected value of the final position $x(t_1)$.
- (b) [15P] Estimate the initial velocity $v(t_0)$ given that we have measured $x(t_1) = 150$.
To account for noise in the measuring device, assume that the measurement itself is actually a Gaussian-distributed random variable with mean 150 and variance 10.

Hints:

- Remember that WEBPPL does not support loops (neither **for** nor **while**) but it does support recursion.
- Use WEBPPL's **observe** statement to condition on the value drawn from a *continuous* distribution being equal to another value.
- In this exercise you can read off your answers from the plots output by the WebPPL interpreter.