1 Synchronisation

Some baggage transportation system of a small airport has two service desks S₁ and S₂ for the baggage drop-off to three different destinations A, B, and C.

Via two flaps F₁ and F₂ (        indicates the closed and        the open position) any baggage can be transferred from S₁ → A ∨ B and from S₂ → B ∨ C – combinations S₁ → C and S₂ → A are not possible. Furthermore, parallel transfer from S₁ ∧ S₂ → B cannot be served in parallel and, hence, is forbidden.

Give a valid synchronisation where P(Fᵢ) indicates to open and V(Fᵢ) to close flap Fᵢ. You can use as many semaphores σᵢ as necessary for which P(σᵢ) and V(σᵢ) are to be operated in the usual way. Discuss if your solution is fair!

2 MPI

Given is the following MPI code fragment (with mod denoting the division with remainder). Each process stores a variable val which is initialised with some integer value. In total, there are always \( p = 2^K \), with \( K \geq 1 \), processes running.

```
val ← rank + 1
off ← 1
for i ← 1 to K do
    tmp ← 2^i
    if (rank mod tmp) = 0 then
        MPI_Recv(buf, 1, MPI_INT, rank + tmp/2, ...)
        val ← val + buf
    else if (rank mod tmp) = off then
        MPI_Send(val, 1, MPI_INT, rank - tmp/2, ...)
    fi
    off ← 2*off
od
```
a) Sketch the communication pattern (using arrows) for \( K = 3 \), i.e. \( p = 8 \) processes (drawn as circles), and write the current value of each process’ variable \( \text{val} \) into the corresponding box for each update step! What is the program computing?

\[
\begin{align*}
\text{initialisation} & \quad \begin{array}{cccccccc}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\end{array} \\
\text{after 1. step} & \quad \begin{array}{cccccccc}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\end{array} \\
\text{after 2. step} & \quad \begin{array}{cccccccc}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\end{array} \\
\text{after 3. step} & \quad \begin{array}{cccccccc}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\end{array}
\end{align*}
\]

b) Compute the speed-up \( S(p) \) and parallel efficiency \( E(p) \) of the above program (communication and updates to variables \( \text{tmp} \) and \( \text{off} \) are to be neglected) and draw a small sketch for both speed-up and efficiency for different values of \( K \)! Discuss your results!

3 Dependence Analysis

Given is the following code fragment.

\[
\text{for } i \leftarrow 1 \text{ to } N \text{ do} \\
\text{for } j \leftarrow 1 \text{ to } M \text{ do} \\
\text{S}_1: \quad A(i+1, j-1) \leftarrow B(i, j) + E(i+1, j+1) \\
\text{S}_2: \quad B(i, j-1) \leftarrow A(i, j) + 1 \\
\text{S}_3: \quad C(2*i+1, j) \leftarrow A(i+1, j-1) + 4 \\
\text{S}_4: \quad D(i, j) \leftarrow C(i, j-1) - 1 \\
\text{od} \\
\text{od}
\]

a) Examine these statements and name all occurring dependencies! Input dependencies may be neglected. Assume matrices \( A, B, C, D, \) and \( E \) have sufficient size and, thus, all index operations are valid.

b) State for each dependency named in (a) both distance and direction vector in order to check if it is a loop-carried or a loop-independent dependency. Could the two nested loops be (partially) run in parallel? Justify your answer!