Task 1: Scheduling

Consider the sequence graph in Figure 1.

All the operations are handled by the same resource type and have the same execution time: 
\( D_+ = D_- = D_\leq = D_\geq = 1 \).

a) Set up a system of inequations which represent the constraints to valid schedules.

b) Set up an optimization model for the optimization of the latency \( L \). Resource constraints need not be taken into account. (Hint: add an objective function to the system of inequations to get a linear program).
c) What is the minimal achievable latency

- if there is only one resource unit which handles all operations?
- if there are unlimited resource units?

Indicate valid starting times for the operations in both cases and check the validity of the schedules by means of the system of inequations determined previously.

System of inequations:

\[
\begin{align*}
\tau(v_3) &\geq \tau(v_1) + 1 \\
\tau(v_3) &\geq \tau(v_2) + 1 \\
\tau(v_4) &\geq \tau(v_3) + 1 \\
\tau(v_5) &\geq \tau(v_4) + 1 \\
\tau(v_5) &\geq \tau(v_7) + 1 \\
\tau(v_7) &\geq \tau(v_6) + 1 \\
\tau(v_9) &\geq \tau(v_8) + 1 \\
\tau(v_{11}) &\geq \tau(v_{10}) + 1 \\
\tau(v_1) &\geq \tau(v_0) \\
\tau(v_2) &\geq \tau(v_0) \\
\tau(v_6) &\geq \tau(v_0) \\
\tau(v_8) &\geq \tau(v_0) \\
\tau(v_{10}) &\geq \tau(v_0) \\
\tau(v_n) &\geq \tau(v_5) + 1 \\
\tau(v_n) &\geq \tau(v_9) + 1 \\
\tau(v_n) &\geq \tau(v_{11}) + 1
\end{align*}
\]

Initial condition:

\[\tau(v_0) = 0\]

Objective function:

\[\min \tau(v_n) - \tau(v_0)\]

Minimum latency and valid starting times:

- One resource: \(L_{\min} = 11\)
- Unlimited resources: \(L_{\min} = 4\)

1 resource: e.g.

\[
\begin{align*}
\tau(v_1) &= 0; \tau(v_2) = 1; \tau(v_3) = 2; \tau(v_4) = 3; \tau(v_5) = 4; \\
\tau(v_7) &= 5; \tau(v_9) = 6; \tau(v_8) = 7; \tau(v_9) = 8; \tau(v_{10}) = 9; \\
\tau(v_{11}) &= 10; \tau(v_n) = L = 11
\end{align*}
\]

Unlimited resources: e.g.
Task 2: Design space exploration

Consider again the sequence graph and the specification of task 1. Assume that there is only one resource type which can compute all operations \((+, -, <, \ast)\) and has an area of 1. The cost of an implementation is given by the total required area. The goal is to find the Pareto-points of the design space which is given by the parameters cost and latency. The number of allocated resources is not yet fixed.

a) Compute a lower and an upper bound for the latency in order to limit the possible Pareto-points.

b) Find a lower and an upper bound for the cost in order to limit the possible Pareto-points.

c) Find all Pareto-points and represent them in a diagram.

Latency bounds: \(4 \leq L \leq 11\). Cost bounds: \(1 \leq c \leq 5\). Pareto-points: see Figure 3.

---

**Figure 2: Starting times of operations for unlimited resources**

**Figure 3: Pareto-points**
Task 3: Marked Graphs

a) Consider the marked graph in Figure 4. The node labeled + represents an addition of the two input values.

I) At the input a a sequence of numbers is read in, with a(k) representing the k-th number. Determine the outgoing sequence b(k) as function of the input values.

II) The initial mark with the value s is replaced by n marks s_1, ..., s_n. Determine a recursive formula for the output sequence b(k).

b) Consider the marked graph in Figure 5. s_1 and s_2 are the data values of the initial marks. The nodes a, b, c and d execute the operations sum, multiplication, forwarding and duplication on the corresponding input values. (Assume that all generated marks have the same value in case of multiple output edges). At the input u a sequence of numbers u(k) is read in. Determine the output sequence v(k) as function of the input values.

\begin{align*}
a) \quad & I) \quad b(1) = a(1) + s \\
& \quad b(k) = a(k) + b(k - 1) \text{ for } k > 1 \\
& \quad \text{with } s \text{ being the data value of the initial mark} \\
& II) \quad \text{For } n = 0 \text{ the output sequence is empty.} \\
& \quad \text{For } n > 0: \\
& \quad b(k) = a(k) + s_k \text{ for } k \leq n \\
& \quad b(k) = a(k) + b(k - n) \text{ for } k > n \\
& \quad \text{with } s_1, \ldots, s_n \text{ being the data values of the initial marks.} \\
\end{align*}

\begin{align*}
b) \quad & v(1) = 2 \ast s_2 \ast (u(1) + s_1) \\
& v(2) = 2 \ast v(1) \ast (u(2) + s_2) \\
& v(k) = 2 \ast v(k - 1) \ast (u(k) + v(k - 2)) \text{ for } k > 2
\end{align*}

Task 4: Marked Graphs

Given the formula of the Fibonacci sequence:

![Figure 4: Marked Graph 1](image)
Fib(n) = \begin{cases} 
0 & \text{if } n = 0 \\
1 & \text{if } n = 1 \\
Fib(n - 1) + Fib(n - 2) & \text{if } n > 1 
\end{cases}

Draw the equivalent marked graph representation (All the initial mark values should be clarified). The desired output sequence should start from \( n = 1 \), i.e., 1, 1, 2, 3, 5, ....

\[ Fib(n) = Fib(n - 1) + Fib(n - 2) \]
\[ x1(n) = Fib(n - 2) \]
\[ x2(n) = x1(n) + Fib(n - 1) \]

See Figure 6.

**Task 5: Sequence graphs**

Represent the following program as sequence graph:
\[ w := a + b \cdot d; \]
\[ x := a - b \cdot e; \]
\[ y := w \cdot c; \]
\[ z := a + b + d; \]
\[
\text{IF} \ (z < 0) \ \text{THEN} \\
\quad z := -z; \\
\text{END IF} \\
\text{IF} \ (y > 0) \ \text{THEN} \\
\quad v := 1 / y; \\
\text{ELSE} \\
\quad v := z \cdot y; \\
\text{END IF} \]