Example: static scheduling

Problem: Assume a system with tasks and precedence constraints according to the figure below. Timing constraints for the tasks are given in the table. Generate a static schedule for these tasks by simulating preemptive earliest-deadline-first scheduling.

![Task Graph]

<table>
<thead>
<tr>
<th>Task</th>
<th>(C_i)</th>
<th>(O_i)</th>
<th>(D_i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>4</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>B</td>
<td>3</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>C</td>
<td>5</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

Period: 15

Begin by calculating the LCM of the tasks: \(\text{LCM}(15, 5) = 15\)

Then generate a new version of the task graph with cycle time 15.

Observe that D must execute \(15/5 = 3\) times within the cycle, hence instances \(D'\) and \(D''\) in the new graph.

Example: static scheduling

Now generate a schedule by assuming preemptive, earliest-deadline-first scheduling and simulate execution of the tasks:

1. A is scheduled first since it has the earliest deadline among the tasks (A, B, C) that are ready at \(t = 0\).
2. D becomes ready at \(t = 3\) and preempts A since D’s deadline is closer in time.
3. A resumes its execution at \(t = 4\) and is finished at \(t = 5\).
4. B is scheduled at \(t = 5\) and is finished at \(t = 8\).
5. D’ becomes ready and is scheduled at \(t = 8\) since the deadline of D’ is closer in time than C’s deadline.
6. C is scheduled at \(t = 9\).
7. D’ becomes ready at \(t = 13\) and preempts C since the deadline of D’ is closer in time.
8. C resumes its execution at \(t = 14\) and is finished at \(t = 15\).

Static schedule:

Cyclic time table:

\((A,0,3) \ (D,3,4) \ (A,4,5) \ (B,5,8) \ (D',8,9) \ (C,9,13) \ (D'',13,14) \ (C,14,15)\)