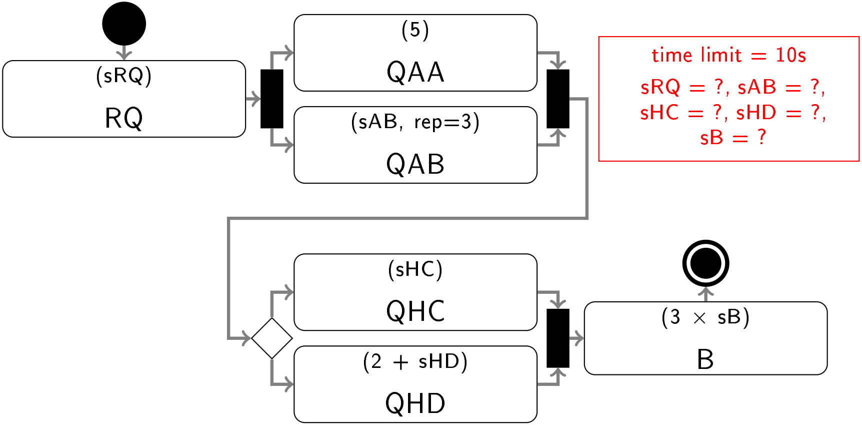
Incremental time limit inference

All paths from the initial node to the final nodes must finish within 10 seconds. We will infer the resulting time limits for each activity.

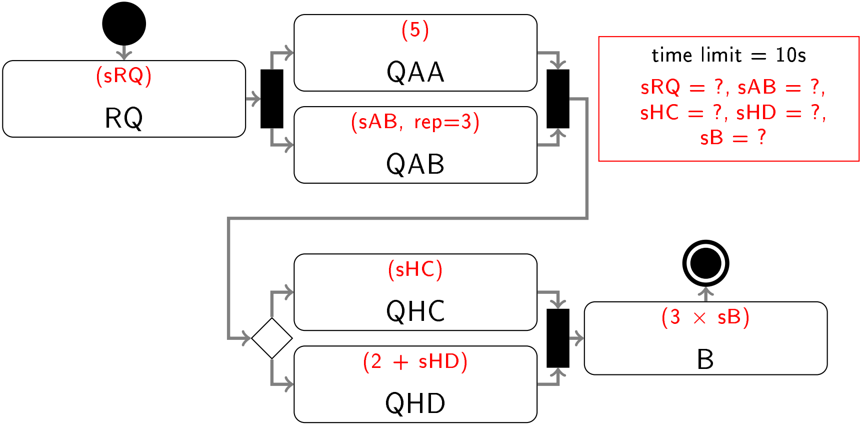


time limit = 10s

sRQ = ?, sAB = ?, sHC = ?, sHD = ?, sB = ?

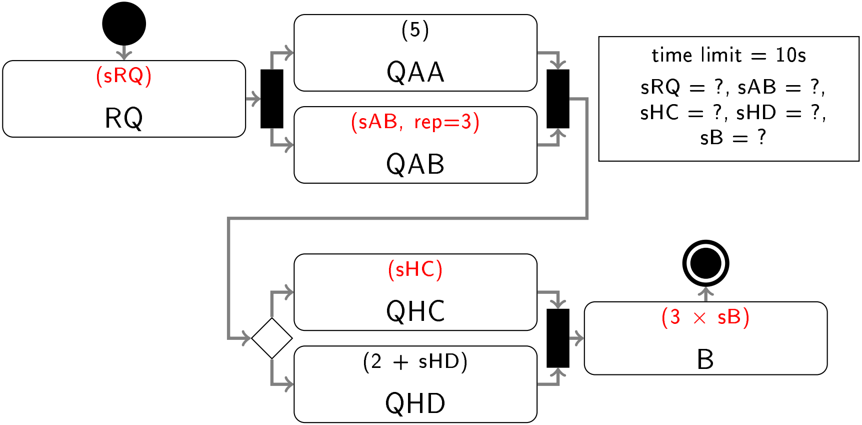
Incremental time limit inference

Activities are annotated with $m + wS$. m is the minimum time limit, w is the weight and S will be the computed slack per unit of weight. Some activities can be repeated multiple times.



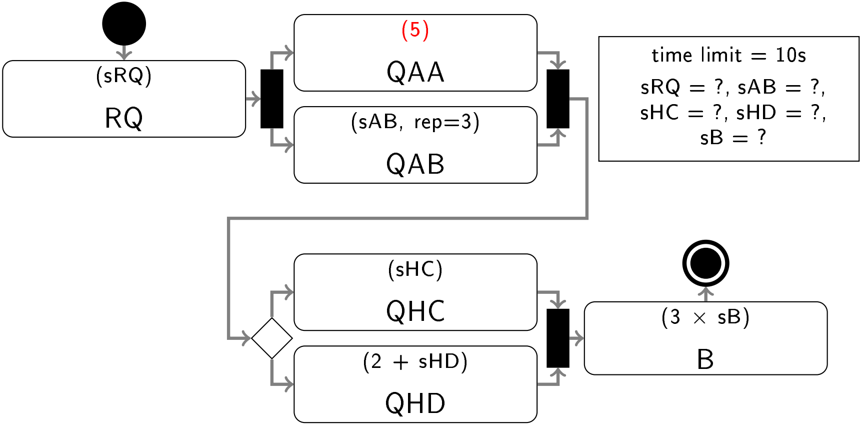
Incremental time limit inference

Most activities will have $m = 0$. w will be an estimation of their relative computational cost.



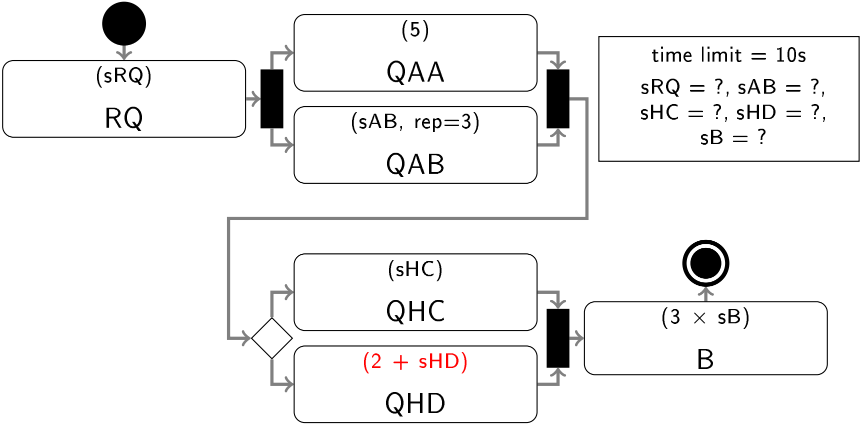
Incremental time limit inference

Other activities may have Service Level Agreements in place, with a previously agreed time limit.



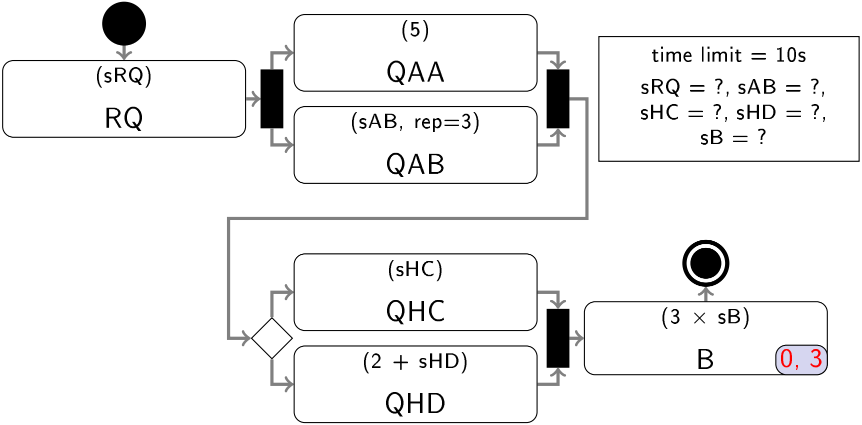
Incremental time limit inference

Finally, some activities may combine a fixed part with the variable part computed by the algorithm.



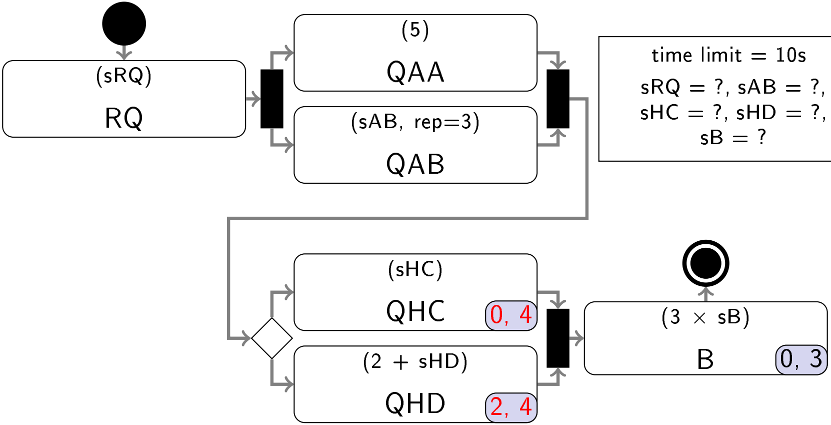
Incremental time limit inference

We need to compute the total minimum time limit and weight of the strictest subpath from each activity. We start from the final node.



Incremental time limit inference

We continue in reverse topological order.

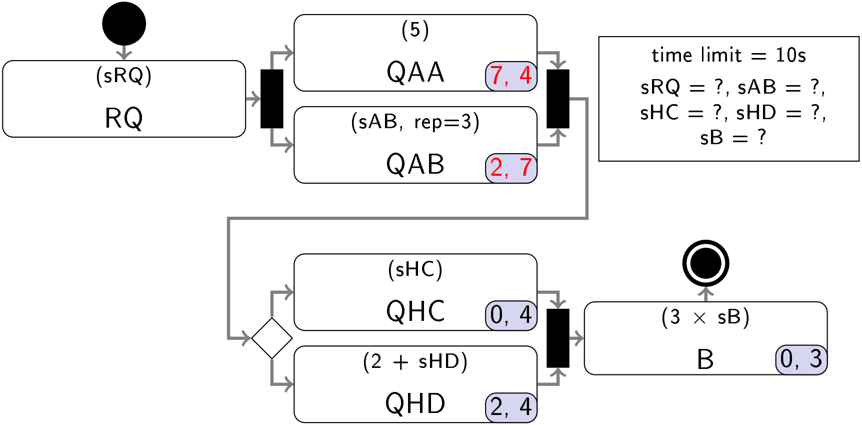


time limit = 10s

sRQ = ?, sAB = ?,
sHC = ?, sHD = ?,
sB = ?

Incremental time limit inference

Between (0, 4) and (2, 4), (2, 4) is always stricter: discard (0, 4) and send (2, 4) back to the join node.

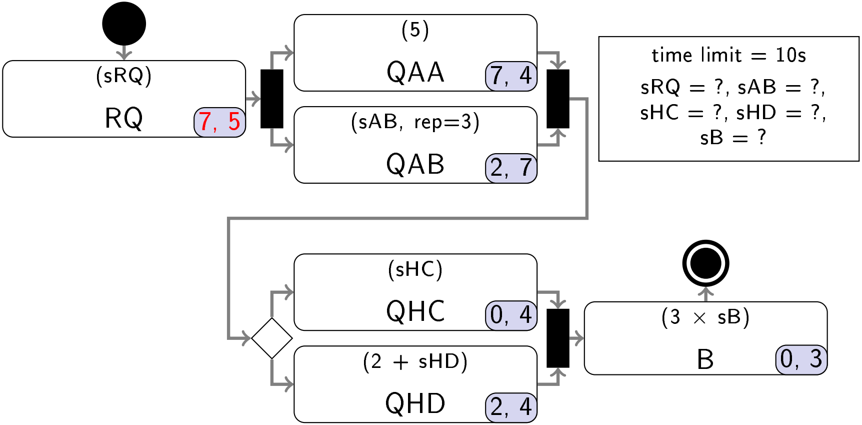


time limit = 10s

sRQ = ?, sAB = ?,
sHC = ?, sHD = ?,
sB = ?

Incremental time limit inference

This time, we propagate (7, 4) back up. The strictest path in the graph from the initial node to a final node is (7, 5).

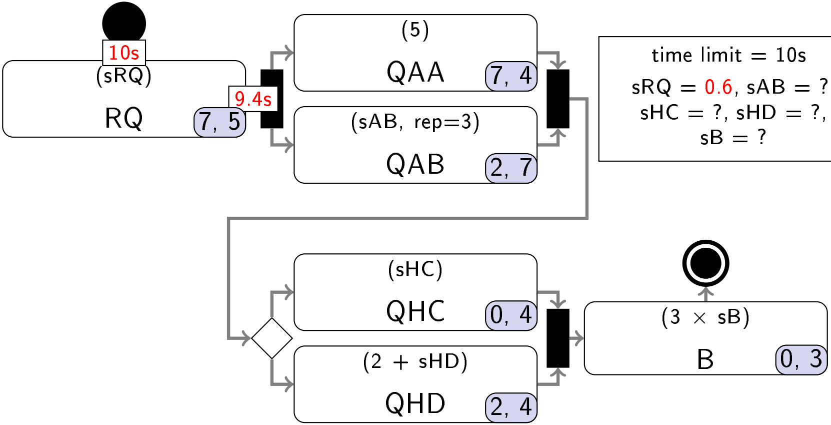


time limit = 10s

sRQ = ?, sAB = ?,
sHC = ?, sHD = ?,
sB = ?

Incremental time limit inference

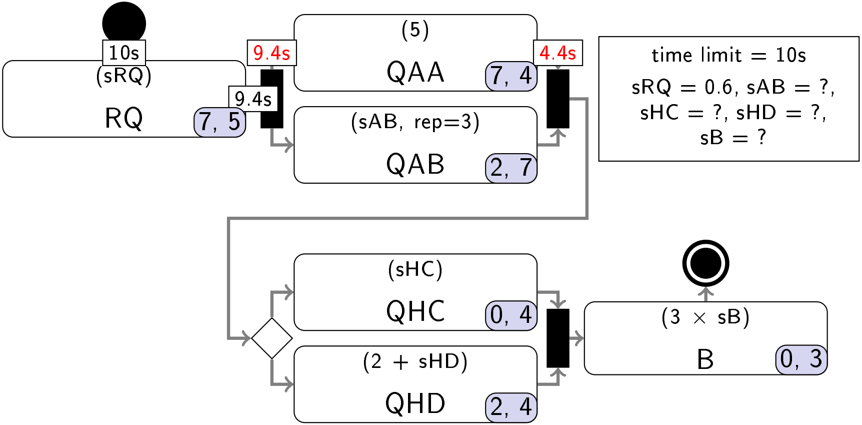
We send 10s into RQ. RQ uses up $0 + 1(10 - 7)/5 = 0.6$s and sends the remaining 9.4s to the fork node.



time limit = 10s
sRQ = 0.6, sAB = ?,
sHC = ?, sHD = ?,
sB = ?

Incremental time limit inference

QAA uses up exactly 5s and sends the remaining 4.4s into the join node.

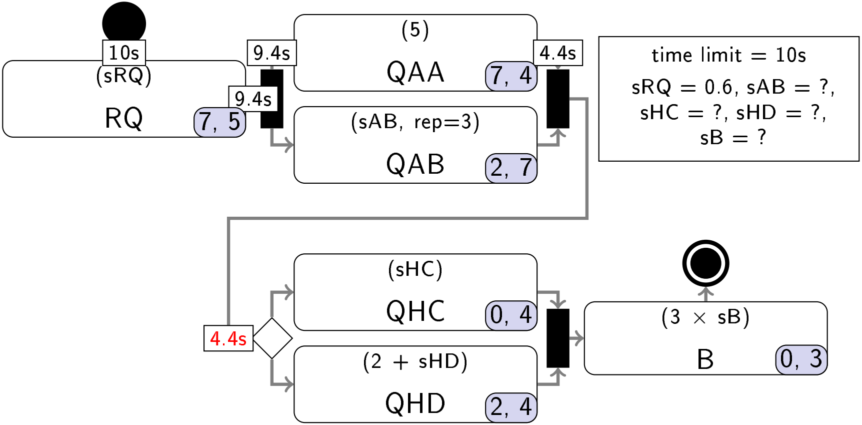


time limit = 10s

sRQ = 0.6, sAB = ?, sHC = ?, sHD = ?, sB = ?

Incremental time limit inference

The join node does not use up any time.

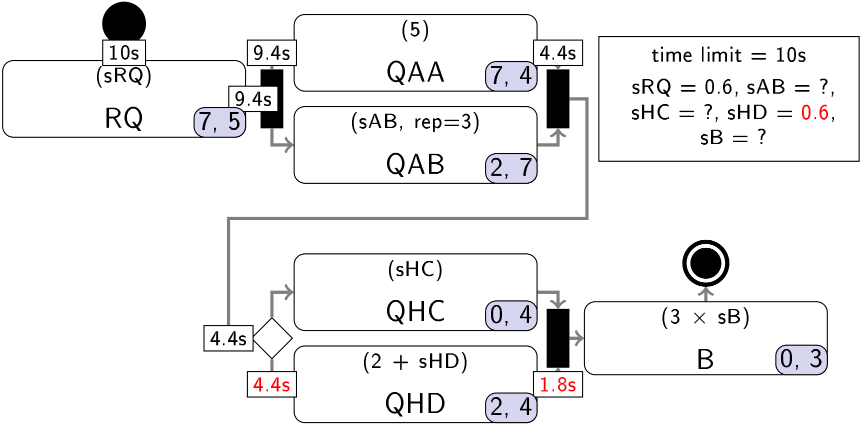


time limit = 10s

sRQ = 0.6, sAB = ?,
sHC = ?, sHD = ?,
sB = ?

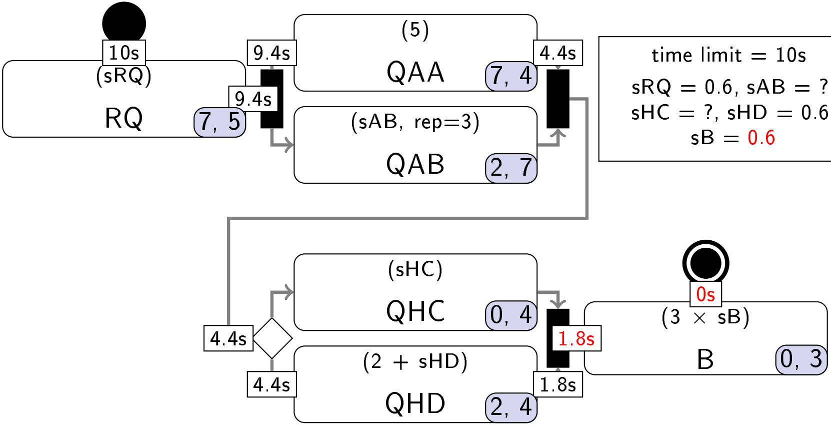
Incremental time limit inference

QHD receives 4.4s, uses up $2 + 1(4.4 - 2)/4 = 2.6$s and sends the rest to the merge node.



Incremental time limit inference

B receives 1.8s and uses up the remaining $0 + 3(1.8 - 0)/3 = 1.8$s.

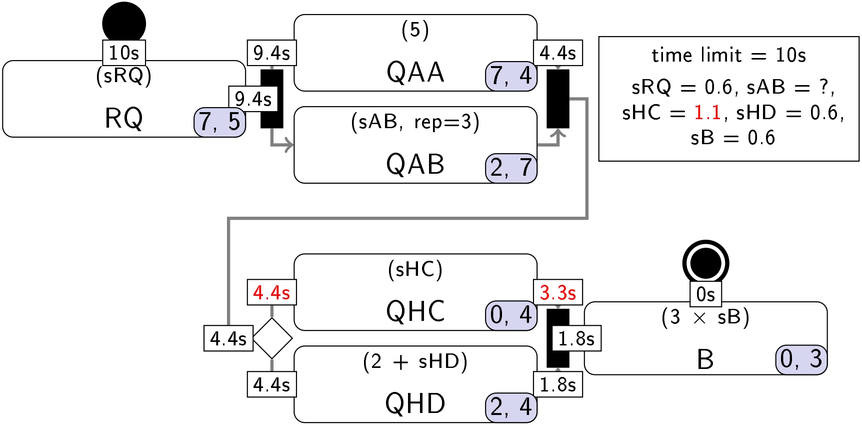


time limit = 10s

sRQ = 0.6, sAB = ?,
sHC = ?, sHD = 0.6,
sB = 0.6

Incremental time limit inference

We back up into QHC. At first, QHC uses up 1.1s and sends 3.3s into the merge node.

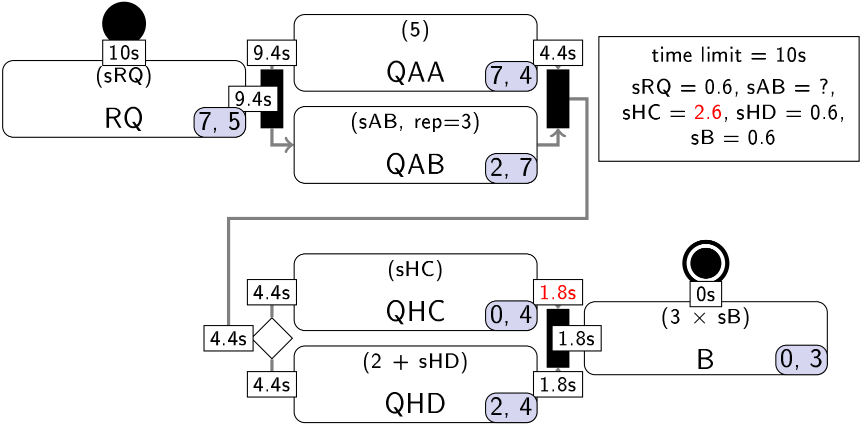


time limit = 10s

sRQ = 0.6, sAB = ?,
sHC = 1.1, sHD = 0.6,
sB = 0.6

Incremental time limit inference

However, the merge node already received 1.8s: we will dedicate the extra 1.5s to QHC.



10s

(sRQ)

RQ

7, 5

9.4s

9.4s

(5)

QAA

7, 4

(sAB, rep=3)

QAB

2, 7

4.4s

time limit = 10s

sRQ = 0.6, sAB = ?,

sHC = 2.6, sHD = 0.6,

sB = 0.6

4.4s

4.4s

4.4s

(sHC)

QHC

0, 4

(2 + sHD)

QHD

2, 4

1.8s

1.8s

1.8s

0s

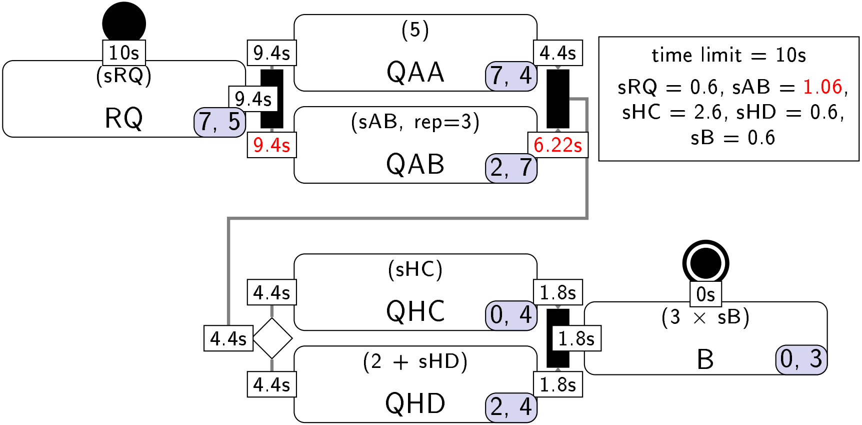
(3 × sB)

B

0, 3

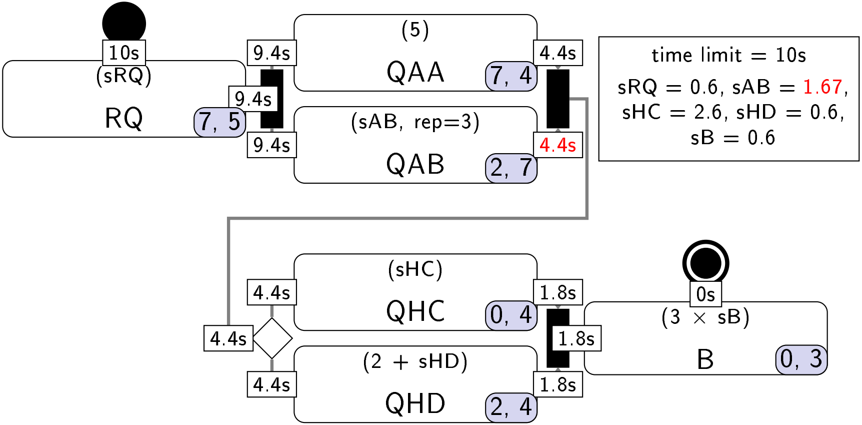
Incremental time limit inference

We back up into QAB. Again, QAB only uses up 3.18s and sends too much time to the merge node.



Incremental time limit inference

We reuse the extra 1.82s into QAB, and we are done.



time limit = 10s

sRQ = 0.6, sAB = 1.67,
sHC = 2.6, sHD = 0.6,
sB = 0.6