**Official solutions and marking scheme for (1a)**

**1ai.** Yes. Every initial path from s1 passes through one of the states s2, s3, and s5 infinitely often, and c is true in all these states. Hence, c is true infinitely often.

**1aii.** Yes. Consider any initial path pi from s1 such that pi |= X(not c). Then, pi[1] = s4. The only states reachable from s4 in one step are s2, s3, and s5, and c is true in all of them. Hence pi[1..infty] |= Xc and therefore pi |= XXc.

**1aiii.** Notice that a is true in s\_1, and then either b or c is true in all of s2, s3, s4.

**Marking scheme:** for each point (i)-(iii), score 0 if point not answered, 1 for wrong answer, 2 for correct answer but with unsufficient [sic] justification, 3 for correct, well-motivated answer. Bonus 1 if scored 3 on all points, for a total of 10 in case of three correct, well-motivated answers.

**Official solutions and marking scheme for (1b)**

**1bi.** False. No matter what robot 1 does (either pushing or waiting), robot 2 has always a way of moving away from pos0 (either waiting or pushing respectively).

**1bii.** True. In q0 atom pos0 is true and robots 1 and 2 have a joint strategy to stay there, either by both pushing or both waiting.

**1biii.** True. In q0 robots 1 and 2 can coordinate to move to q1 (e.g., robot 1 pushes and robot 2 waits) and then they can stay there as illustrated in point (ii)

**Marking scheme:** for each point (i)-(iii), score 0 if point not answered, 1 for wrong answer, 2 for correct answer but with unsufficient [sic] justification, 3 for correct, well-motivated answer. Bonus 1 if scored 3 on all points, for a total of 10 in case of three correct, well-motivated answers.

**Official solutions and marking scheme for (2a)**

**2ai.** No. In fact, s0 |=/= Phi1, as a is false in s0.

**2aii.** True. [sic] We have that s4 |= AGb, as b is true in s4 and no other state is reachable from s4. Since s4 is reachable in one step from s0, we have that s0 |= EX(AGb), and therefore s0 |= Phi2.

**2aiii.** No, as a is false in s0.

**Marking scheme:** for each point (i)-(iii), score 0 if point not answered, 1 for wrong answer, 2 for correct answer but with unsufficient [sic] justification, 3 for correct, well-motivated answer. Bonus 1 if scored 3 on all points, for a total of 10 in case of three correct, well-motivated answers.

**Official solutions and marking scheme for (2b)**

**2bi.** False. No matter what robot 1 does (either pushing or waiting), there is always an execution leading away from pos0 (where robot 2 either waits or pushes respectively).

**2bii.** True. In q0 atom pos0 is true and robots 1 and 2 have a joint strategy to stay there, either by both pushing or both waiting.

**2biii.** True. In q0 robots 1 and 2 can coordinate to move to q1: robot 1 pushes and robot 2 waits. Then, they can remain in q1 if they both wait. In particular, this is a uniform strategy for robot 2, who cannot distinguish between q0 and q1.

**Marking scheme:** for each point (i)-(iii), score 0 if point not answered, 1 for wrong answer, 2 for correct answer but with unsufficient [sic] justification, 3 for correct, well-motivated answer. Bonus 1 if scored 3 on all points, for a total of 10 in case of three correct, well-motivated answers.

**Official solutions and marking scheme for (3a)**

Yes. Agent 1 and 2 can first move from state 12 to state 8 by loading the cargo (agent 1 loads and agent 2 does nop)

Then, agent 1 can fly the rocket to London (state 5), no matter what the other agents do.

Finally, agents 1 and 2 can unload the cargo, thus reaching state 2, where atom caL is true.

**Marking scheme:** score 0 if question not answered, 3 for question attempted but wrong answer, 6 for correct answer but with unsufficient [sic] justification, 10 for correct, fully-motivated answer.

**Official solutions and marking scheme for (3b)**

The model check algorithm starts with the set Sat(caL) = {1, 2, 3, 4} of states that satisfy caL, and then iterates to compute the states from which Sat(caL) is reachable in 1, 2, … steps, until no new state is added.

In our case, we obtain the following iteration:

S0 = {1, 2, 3, 4} = Sat(caL)

S1 = {1, 2, 3, 4, 5, 6}

S2 = {1, 2, 3, 4, 5, 6, 8}

S3 = {1, 2, 3, 4, 5, 6, 8, 12}

S4 = {1, 2, 3, 4, 5, 6, 8, 12}

Since S3 = S4, the algorithm stops and returns Sat(phi) = S3.

**Marking scheme:** score 0 if question not answered, 3 for question attempted but wrong answer, 6 for correct answer but with unsufficient [sic] justification, 10 for correct, fully-motivated answer.