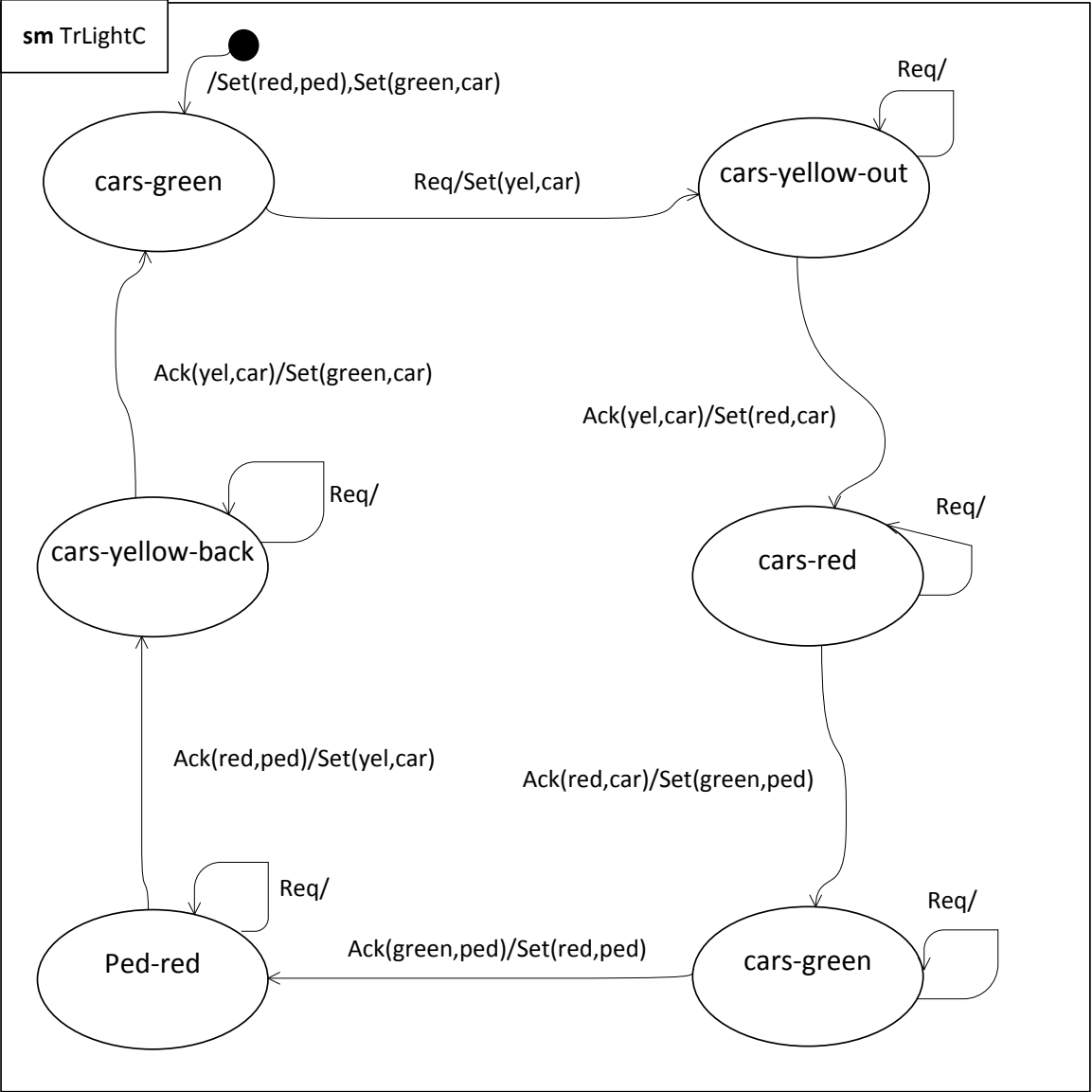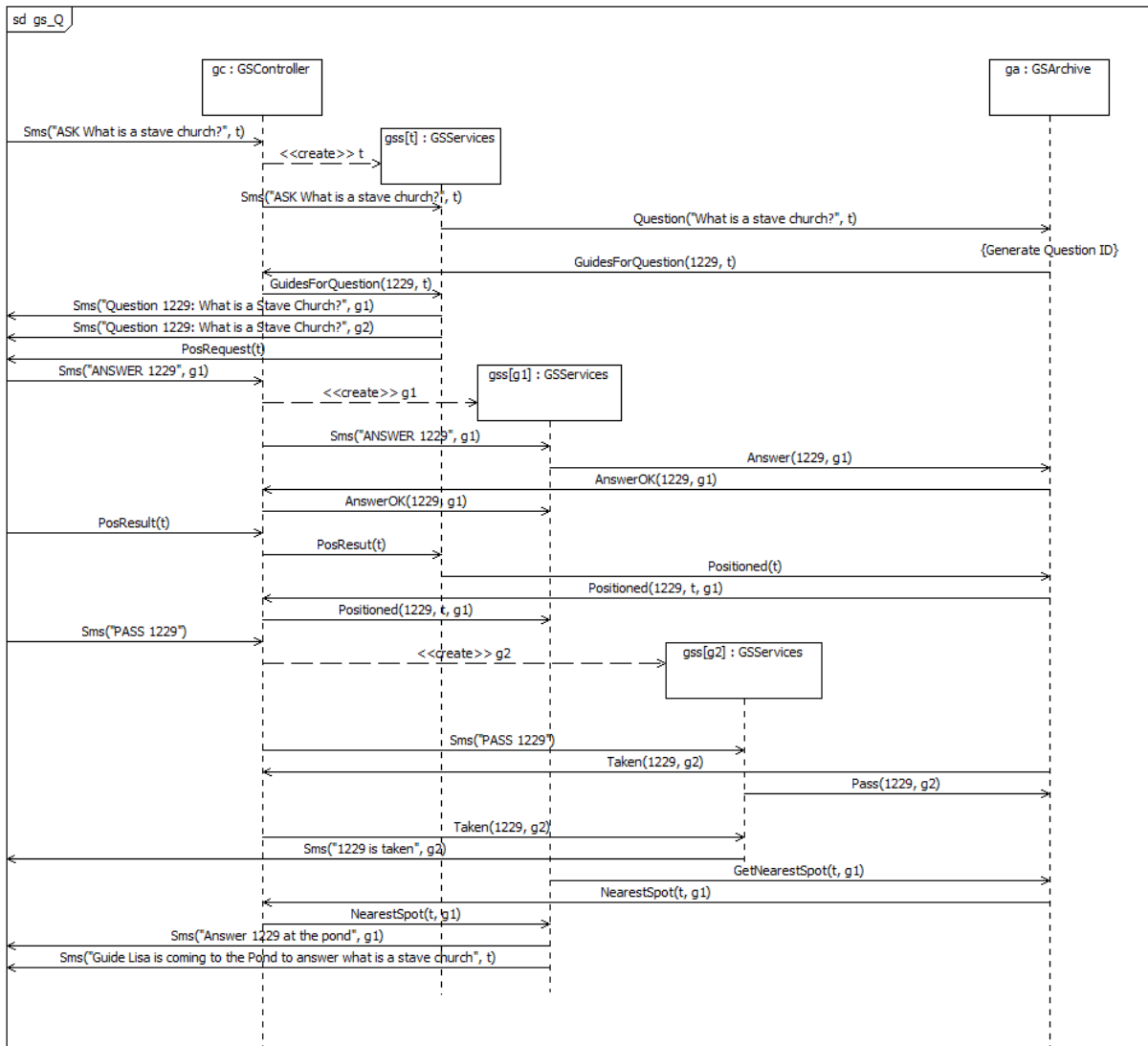# Solution

## 1)

## 2a)



Remarks: last parameter in signals is always the routing parameter (identifier). Lifelines that explicitly end within the Interaction are *destroyed/terminated*.

Points to make here:

1. The gates match the events on the lifeline precisely

2. Every time an Sms enters from the environment a new lifeline is created (3 times)

3. gc serves as a router such that all communication internally must go through the gc

4. External communication from gss should go directly out

## 2b)

The state machine in Figure 3 is consistent with the sequence diagram in Figure 1. When the sequence diagram has been exhausted, the state machine is in state Answering.

## 2c)

The state machine is consistent until state Answering. Upon reception of another Sms with command == Question it will consume the Sms and not reply with either PASS or ANSWER. Thus the state machine is not very good for the repetitive case.

Remedies to improve this would be:

1. Use a timer to control when to leave Answering. This means that the answer takes a given amount of time. Hence, the transition for leaving Answering could for example be decorated with `tm(15 minutes)/`. The idea is that the guide should finish talking to the tourist in less than 15 minutes. One should also defer the Sms signal in the Answering state. This means the state Answering should have a transition to itself with the following decoration: `sms/defer`.

2. Alternatively, it is technically possible not to have the Answering state, and only have the Walkabout state. In reality this means that any new request will have priority over the execution of answering an already accepted request. Thus only one state is not really recommended. Give less that full score.

3. Yet another and good alternative is to introduce active leaving the Answering state. In reality this could mean that the Guide actually clicks on some button to accept the (new) Sms-es. Technically, a new signal (e.g. "ready") could be introduced which should take the place of the Sms on the transition from Answering to Walkabout. It would also be needed to defer Sms in Answering.

## 2d)

Four.

## 2e)

One. The first event of lifeline t.

## 2f)

Three. The last events of t, g1 and g2.

## 2g)

In order to increase the number of potential first events we add the two red message arrows. To get only two possible last events we add the green message arrow.

**sd Q**

| t:Tourist | g1:Guide | g2:Guide | gs:GuideSystem |
|---|---|---|---|

a →

b →

Sms("ASK What is a stave church?", gs) →

Sms("Question 1229: What is a Stave Church", g1) ←

Sms("Question 1229: What is a Stave Church", g2) ←

PosRequest("1229",t) ←

Sms("ANSWER 1229", gs) →

PosResult("1229", gs) →

Sms("PASS 1229", gs) →

Sms("1229 is taken",g2) ←

c →

Sms("Answer 1229 at the pond",g1) ←

Sms("Guide Lisa is coming to the Pond to answer what is a stave church",t) ←