

Assignment 6: Model Checking of Real-Time Systems

Goal: Design a communication protocol, where the communication between the **Sender** and the **Receiver** is done through a lossy and noisy **Channel**.

Description of the function: The **Sender** sends the message, and awaits the acknowledgment from the **Receiver**. Since the communication is done through the **Channel**, the message might be passed to the **Receiver** after 1 time unit or lost. If the **Receiver** receives the message, it will send the acknowledgement after 1 time unit. Again, this is done through the **Channel**, so the acknowledgement can be sent to the **Sender** after 1 time unit or lost. In case the acknowledgement does not reach the **Sender** in 5 time units (either the message or the acknowledgement is lost), the **Sender** will send the message again without delay. Similarly, after the acknowledgement is received, the **Sender** will send another message without delay.

Your job: Model the system in UPPAAL and verify the following properties:

- The system cannot deadlock;
- The receiver might send an acknowledgment;
- The messages sent by the sender are eventually received by the receiver.

Note: You are allowed to use ChatGPT for helping you building the model. However, please note that ChatGPT's answer may not be completely right. You must analyze by yourself and selectively use the answers. You may need to use prompts to tune your ChatGPT first. Please consider using the following example of a prompt.

Input the following text into ChatGPT:

Example: A UPPAAL can look like this:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE nta PUBLIC "-//Uppaal Team//DTD Flat System 1.6//EN" 'http://www.it.uu.se/research/group/darts/uppaal/flat-1_6.dtd'>
<nta>
  <declaration>chan a,b;
  int var = 5;</declaration>
  <template>
    <name x="5" y="5">TemplateA</name>
    <declaration>clock timer;</declaration>
    <location id="id0" x="0" y="0">
      <name x="10" y="34">L0</name>
    </location>
    <init ref="id0"/>
    <transition id="id1">
      <source ref="id0"/>
      <target ref="id0"/>
      <label kind="select" x="41" y="102">i:int{0,2}</label>
      <label kind="guard" x="41" y="85">timer>10</label>
      <label kind="synchronisation" x="41" y="68">a!</label>
      <label kind="assignment" x="41" y="51">timer:=0</label>
```

```

                                <nail x="-59" y="-51"/>
                                <nail x="51" y="51"/>
                                </transition>
</template>
<template>
    <name>TemplateB</name>
    <parameter>int para1, int para2</parameter>
    <location id="id2" x="-204" y="0">
        <name x="-214" y="-34">L0</name>
    </location>
    <location id="id3" x="-51" y="0">
        <name x="-61" y="-34">L1</name>
        <committed/>
    </location>
    <location id="id4" x="-127" y="93">
        <name x="-137" y="59">L2</name>
        <urgent/>
    </location>
    <location id="id5" x="-357" y="0">
        <name x="-367" y="-34">L3</name>
    </location>
    <init ref="id2"/>
    <transition id="id6">
        <source ref="id2"/>
        <target ref="id5"/>
        <label kind="synchronisation" x="-262" y="4">b?</label>
        <nail x="-280" y="42"/>
    </transition>
    <transition id="id7">
        <source ref="id5"/>
        <target ref="id2"/>
        <nail x="-280" y="-42"/>
    </transition>
    <transition id="id8">
        <source ref="id4"/>
        <target ref="id2"/>
    </transition>
    <transition id="id9">
        <source ref="id3"/>
        <target ref="id4"/>
    </transition>
    <transition id="id10">
        <source ref="id2"/>
        <target ref="id3"/>
        <label kind="synchronisation" x="-186" y="17">a?</label>
    </transition>
</template>
<system>
    modelA = TemplateA();
    modelB = TemplateB(1,2);
    system modelA, modelB;
</system>

<queries>
    <A suggestion of a question to ChatGPT>
        <formula>E</formula>
        <comment>Reachability</comment>
    </result>
    </A suggestion of a question to ChatGPT>
    <A suggestion of a question to ChatGPT>
        <formula>A</formula>
        <comment>Liveness</comment>
    </result>
    </A suggestion of a question to ChatGPT>
    <A suggestion of a question to ChatGPT>
        <formula>A[] !modelB.L3</formula>
        <comment>Invariance</comment>
    </A suggestion of a question to ChatGPT>
    <A suggestion of a question to ChatGPT>
        <formula>E[] var != 0</formula>
        <comment>
    </A suggestion of a question to ChatGPT>
</queries>
</nta>

```

Requirement: Create a new UPPAAL model of a communication protocol that has a **Sender**, a **Receiver**, and a lossy and noisy **Medium**.

A suggestion of a question to ChatGPT: How many timed automata should I build?

Requirement: The **Sender** sends the message and awaits the acknowledgment from the **Receiver**. Since the communication is done through the **Medium**, the message might be passed to the **Receiver** after 1 time unit or lost.

A suggestion of a question to ChatGPT: How to model the behavior of the Sender sending messages?

A suggestion of a question to ChatGPT: How to model the phenomenon that the message from the Sender might be passed by the Medium to the Receiver after at most 1 time unit?

A suggestion of a question to ChatGPT: How to model the phenomenon that the message from the Sender might be lost by the Medium?

Requirement: If the **Receiver** receives the message, it will send the acknowledgement after 1 time unit.

A suggestion of a question to ChatGPT: How to model the behavior of the Receiver receiving a message?

Requirement: The acknowledgement is sent through the **Medium**, so the acknowledgement can be sent to the **Sender** after 1 time unit or lost.

A suggestion of a question to ChatGPT: How to differentiate between the acknowledgement sent by the sender and the one sent by the medium in the model?

Requirement: In case the acknowledgement does not reach the **Sender** in 5 time units (either the message or the acknowledgement is lost), the **Sender** will send the message again without delay. Similarly, after the acknowledgement is received, the **Sender** will send another message without delay.

A suggestion of a question to ChatGPT: How to model the behavior of the Sender immediately resending a message after 5 time units if it does not receive the reply?

Requirement: Formulate the following temporal logic properties:

- The system cannot deadlock;

.....

If you have used ChatGPT for helping your build the model, **please answer the survey:** <https://forms.gle/yYUWFyqx1BAdbhaF8>. It will take you less than 5 minutes. Your answers will only be used in scientific research and it is anonymous.

Submit the UPPAAL model (.xml file) together with a short report. If you use ChatGPT, please copy and paste your conversation with ChatGPT in a separated report.

If you have any question, welcome to contact Rong Gu, rong.gu@mdu.se.