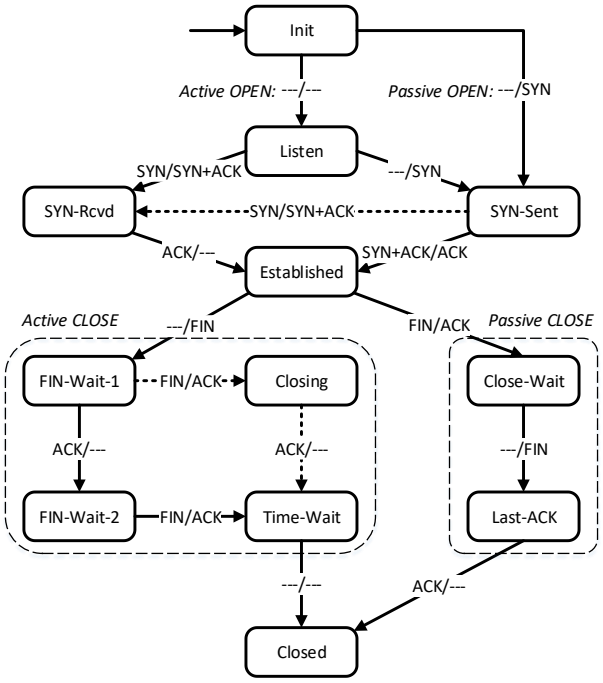
Home Assignment (Model checking)

Title: **TCP**

Advisor: **Ákos Hajdu**

Problem Description

The figure on the right shows the simplified state machine model of the connection establishment layer of the TCP protocol, focusing only on the messages exchanged through the network[1]. The TCP protocol implements a *three-phase handshake* both for the establishment and the termination of connections, negotiating the initial sequence number to be used.



In this task, we *will not model* one of the most important aspects of TCP: the sequence numbers. The goal is to analyze the state machine in terms of the types of exchanged messages. The state machine on the right does not contain transitions that facilitate the early termination of the connection. However, the transitions that serve to avoid race conditions (such as when both participants try to connect actively) are included in the model, denoted by a dashed arrow. Our purpose with this model is to assess the importance of sequence numbers – assuming normal operation, what situations require them and to what extent.

Your task is to model two (identical) participants based on the presented statechart. Network communication – contrary to the channel synchronization of UPPAAL – is asynchronous, so it shall be modelled with message queues. We want to analyze the interactions of two participants, connected with a peer-to-peer FIFO channel for each direction. The size of the channel buffer must be sufficiently large to avoid overflows during operation. There are five types of messages: *SYN, FIN, ACK, SYN+ACK* and *FIN+ACK* (the last two are considered separate messages because they are sent in a single package, therefore they arrive at the same time). The labels of the transitions in the figure are of the format *<received message>/<sent message>*. If the received message is --- then the transition is spontaneous (i.e. it does not require an incoming message to execute), while --- on the right of the label denotes the case where no message is sent.

Hints: The main challenge of the task is the modeling of the FIFO message queues, the statechart should be modelled without modification. To model the messages, it is recommended to define a custom type. Then, the message queue can be an array (for each participant), while sending, checking and reading messages can be implemented in UPPAAL functions. To determine the required size of the channel buffers, a variable may be used to indicate an overflow, so that the proper size can be ensured by model checking. The model should not contain timing and the passing of time during simulation and model checking should be prevented by the proper settings of the states.

[1] A more detailed model can be found here: http://www.medianet.kent.edu/techreports/TR2005-07-22-tcp-EFSM.pdf

Requirements to check

Prove the satisfiability of the requirements below using temporal logic expressions and model checking (in case of unsatisfiability explain the reasons in detail with a counterexample)! Show and explain a short example/counterexample where possible!

1. Both participants can reach the *Established* state simultaneously (i.e. the connection can be established).
2. Both participants can reach the *Closed* state simultaneously (i.e. the connection can be terminated).
3. If the connection gets established, it will eventually get closed, i.e. both participants will reach the *Closed* state.
4. A deadlock can occur only if both participants have closed the connections (i.e. the protocol finished). *Optional task: Can we guarantee this requirement with sequence numbers on the messages? In this special case, the effect of sequence numbers can be simulated by erasing every message from the message queue when a SYN or SYN+ACK message arrives (which are normally the first messages to receive).*