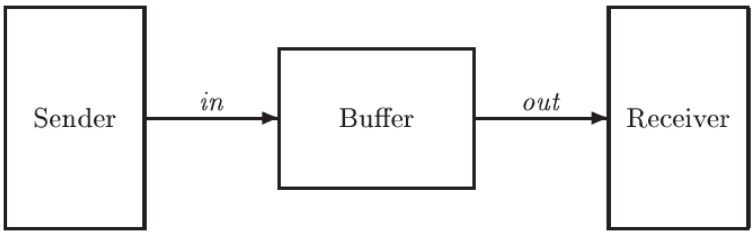
BCS2213 – Formal methods

Teaching assignment 5. TLA specification of the FIFO Protocol.

1. Model of the FIFO (First In First Out) protocol.



- Sender and Receiver interact by messages passing (like in Async Interface);
- they communicate not directly, but via *Buffer*;
- Buffer has a FIFO inside, storing a *finite* number of messages.

2. Specification of FIFO.

- specification of FIFO **Extends** modules **Naturals** and **Sequences**.
- the **Sequences** module defines operations on finite sequences (tuples).
- TLA tuple is represented in ASCII with << >>
- the basic operations on the sequences are:

$Seq(S)$ The set of all sequences of elements of the set S.

$Head(s)$ The first element of sequence s.

For example, $Head(\langle 3, 7 \rangle)$ equals 3.

$Tail(s)$ The tail of sequence s.

For example, $Tail(\langle 3, 7 \rangle)$ equals $\langle 7 \rangle$.

$Append(s, e)$ The sequence obtained by appending element e to the tail of sequence s.

For example, $Append(\langle 3, 7 \rangle, 3)$ equals $\langle 3, 7, 3 \rangle$.

3. Run TLA+ Toolbox and create new module with name lab_5_<your_ID>.tla

4. Define as a tuple variable *Buf* and apply to it all the considered above operations on sequences.

In order TLA module will be correct don't forget to define initial, next state predicates and combining it specification.

5. To understand the sense of the operations on sequences print the resulted *Buf* (for it, extend your module by TLC) .

6. Write TLA specification of FIFO protocol.

Send action appends a message to the *Buf*, ***Rcv*** action deletes a message from the *Buf*.

For implementation, you also need define the constant ***Data***.

7. You have specified an unbounded FIFO, which can hold any number of messages.

But any real system has a finite amount of resources, so FIFO can contain only a bounded number of messages. So action ***Send*** is enabled if there are fewer than *N* messages in the buffer, i.e. Len(*Buf*) is less than N, where N is some constant.

8. Specification of the *bounded* FIFO was its *safety* property.

Define a *liveness* property: a Buffer is eventually often full or eventually often empty.

9. Specify any other *safety* property for FIFO and check it with TLC.

10. Specify any other *livness* property for FIFO and check it with TLC.

11. Don't forget write comments in TLA module, explaining your ideas.

12. Please upload your labsheet into Kalam. It will be evaluated in max 2.5% of your general marks.