A buffer is a storage structure that can be used to hold data that have been constructed and returned by one procedure (the “producer”) and will subsequently be needed as inputs to another procedure (the “consumer”), in situations where the consumer and the producer are not perfectly synchronized. The producer transmits a datum to the buffer when it has been constructed. The consumer extracts a datum from the buffer when it is ready to begin processing that datum.

Buffers usually have finite capacity and so can be full, in which case it is an error for a producer to try to add another datum. It is also an error for a consumer to try to extract a datum when the buffer is empty, having already delivered to the consumer every datum that it has so far received from the producer.

A buffer is responsible for preserving the order of the data that it receives from the producer, so that the order in which the consumer extracts them from the buffer is the same as the order in which the producer transmits them to the buffer.

The basic operations on buffers, then, are creating an empty buffer; checking whether a given buffer is empty; checking whether it is full; transmitting a datum to a non-full buffer (yielding a new buffer that holds the new datum in addition to any held by the given buffer); and extracting a datum from a non-empty buffer (yielding the extracted datum and a new buffer that no longer holds that datum).

The assignment is to design, define, implement, and test a Buffer datatype, first as an abstract data type, then as a collection of Scheme procedures.

This assignment will be due on Friday, September 11. Please submit your solution in hard copy at the beginning of that day’s class session.