

Concurrent ML

Exercise 3: Buffers with Fixed Maximum Lengths, Programmed Without Selective Communication

1 Goal

Without using selective communication (`select` or `choose`, which we'll discuss on Monday), implement buffers with arbitrary, but fixed, maximum lengths.

Your signature should be:

```
signature BUFFER =
sig

  (* a buffer with fixed maximum length whose elements have type 'a *)

  type 'a buffer

  (* exception raised by make *)

  exception Size

  (* create a buffer with a given maximum length; raises Size if
     the length is not positive *)

  val make : int -> 'a buffer

  (* add(buf, x) adds x to the end of buf, and then returns (); it
     blocks until the addition is possible *)

  val add : 'a buffer * 'a -> unit

  (* del buf deletes the first element of buf, and returns the deleted
     element; it blocks until the deletion is possible *)

  val del : 'a buffer -> 'a

end;
```

2 Submission

Bring a copy of your program, or as much of it as you are able to write, to Friday afternoon's exercise session, and also make it available on the WWW. Be prepared to talk about your

solution during the exercise session.