02158 CONCURRENT PROGRAMMING FALL 2024

Exercise Class 4

Thursday November 14

Synchronous Message Passing (CSP)

- 1. Write a CSP process Merge that interleaves two streams of integers coming from processes A and B respectively and passes on the interleaved stream to a process C.
- 2. Write a CSP process Sum that repeatedly receives an integer value from each of two processes A and B (in either order), and passes their sum to a process C.
- **3.** Show how to implement a barrier for three processes using CSP communication (passing void messages with statements of the form P!() and P?()).

Rendezvous

4. Do Andrews Ex. 8.14. Use rendezvous (**in**) to implement the operations using a serverbased module with a structure like:

```
module Account
op deposit(amount : posinteger);
op withdraw(amount : posinteger);
body
var bal : integer := 0;
process AccountServer;
repeat
in deposit(amount) \rightarrow \dots
[] \dots
ni
forever;
```

- end Account;
- 5. Do Andrews Ex. 8.15 using rendezvous. Operations MeetA() and MeetB() should be used by A and B processes respectively for the meeting.
- 6. Do Rendez.1. (In [Aux] page 11.)