Colloquium 5 - Concurrent Programming

Simon Hasir – 7006072

May 30, 2022

---

1 TE-1

(a) No, it's a destructive channel, only one reciever but inititially it's *"send* to all

(b) No, it runs the first case with a matching condition and chooses the defaults non deterministically

(c) No, intchan1 is asynchronous and doesn't wait for a receiver whereas intchan only continues with execution when the massage is fetched by another agent.

(d) No, when every agent terminates

(e) Yes, if you fetch from a sync channel without any input

2 TE-2

(a) return; oder mainAgent();

(b) void x() {...}

(c)
```
mainAgent {
        select {
            default: {
                P
            }
            default: {
                P
            }
        }
}
```

(d) $\alpha; P;$

(e)
```
mainAgent {
        start(P());
        start(Q());
}
```

3 TE-3

(a)
```
    mainAgent {
    stringchan toastchan;
    boolchan ready, go;

    // A toaster with two slots..
    start(slot(toastchan, ready, go));
    start(slot(toastchan, ready, go));
    // ..and two users
    start(user("Conny", toastchan));
    start(user("Dieter", toastchan));

    for(int i = 0; i < 2; i++){
        <? ready;
    }
    for(int i = 0; i < 2; i++){
        go <! true;
    }

}

void slot(stringchan toastchan, boolchan ready, boolchan go) {
    ready <! true;
    <? go; // wait for go before toasting
    string toast = <? toastchan;
    toastchan <! "Crispy " + toast;
}
```

(b) After l. 24, l. 25 can be run instead of slots(). Solution: another stringchan. toast = ¡? toastchan2 l.19 toastchan2 ¡!

4 TE-4

1. fibonacci sequence

2. ```
   while (run) {
       select {
           case c <! v: {}
           case u = <? c: {
               int i = <? k;
               if (i == n) { //terminate and send term signal
                   run = false;
                   b <! true;
                   println(v);
               }
               else {
                   k <! i+1; //inc counter
                   v = v + u;
                   c <! v;
               }
           }
           case <?b: {
               run = false;
           }
       };
   }
   ```
   ```