Various Complex (?) Scenarios for HW2
...and How to Handle Them

Deniz Sayın

Contents

Note: *Agent* refers to both proper privates and sneaky smokers.

1 Blocked Privates Blocking Other Privates

Consider a simple situation in a 4×4 grid where a single proper private P0 is gathering cigarettes from a 2×2 area at (2, 0), shown in Figure 1.

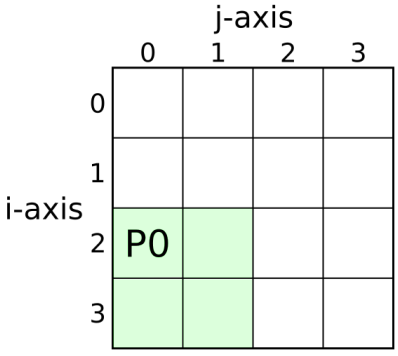

Figure 1: P0 peacefully gathering cigbutts

Then, P1 arrives and wants to gather from a 2×2 area at (1, 1). P1 is blocked because the area is intersecting with P0's area, as shown in Figure 2. So far so good.

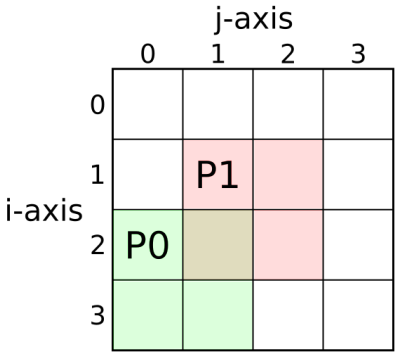

Figure 2: P1 blocked by P0

Finally, P2 arrives and wants to gather from a 2×2 area at (0, 2), as shown in Figure 3. Now: should P2 be blocked or not?

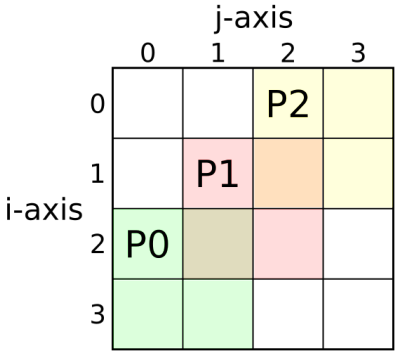

Figure 3: P2 arrives, what now?

Ideally, P2 should of course not be blocked since no one is gathering from the area he wants to

gather from. So, your program should not be blocking P2 in this case. Think about it; a certain type of solution is weak against this situation.

But, in practice, to keep things a bit simpler, I will allow for your implementation to block P2 in this case for **Part I and Part II**. You can get full points from those parts even if you block P2 (including similar cases where privates block others).

I still expect you to implement this properly and not block P2 in this case in your **Part III** implementation. The situation is similar when mixing smokers and privates: agents should not be blocked unless someone is **actively** using an area. I will count blocking P2 and similar cases as an inconsistency when evaluating Part III.

2 Orders in Quick Succession

What happens if there is too short a time between consecutive orders? e.g.

```
...
break 2000
continue 2001
...
```

Obviously, 1 ms is not enough for the order to be executed with any proper amount of agents. If you issue the next order immediately before waiting for everyone to react to the first order, chaos will probably ensue in your program.

So, in this case, your program should wait for the first order to be completed (i.e. everyone has taken the break) before issuing the next order. I do not intend to test such cases a lot, but **there will be a few cases involving quick successive orders for sure**.

Be careful to not block yourself in case you are issuing orders when there are no agents left!

3 Orders Before Initialization is Complete

What if you have to give orders very early in the program, possibly before you have even managed to start every thread? e.g.

```
...
break 2
...
```

This is a problem, especially if there are other orders soon after. **I do not intend to test this**. I am planning on always waiting a few hundred ms before giving orders to avoid such cases, although the exact numbers will depend. Just don't take forever while setting up!

If you want to have a rock-solid implementation, you can wait for every thread to have started before giving your first order. But as I said, this will not be tested as to not further complicate your program.

4 Receiving a BREAK! Order When About to Leave an Area

Let's say a proper private has finished cleaning an area and is about to leave it when he gets the order to take a break. How to approach this?

1. Should the private come back to the area after the break? This could mean more cleaning if someone smoked there in between, or instantly locking-unlocking if no-one did.
2. Should the private move on to the next area on return, thus not having sent a "cleared area" notification for the area they cleared?
3. Or, is it possible to avoid sending the order until the private unlocks the area and sends the "cleared area" notification if your area locking/unlocking process is fast?

All three approaches are acceptable.