Street Simulation

<u>Criteria #1:</u> No more than 3 cars should be present on the street at any instance; let us say it is illegal. Subsequent cars can only wait until any one of the 3 cars leaves.

Logic: *The logic that satisfy this criteria is that in the when* `cars_on_street` *is equal to 3 the it puts the* `incoming_enter()` *and* `outgoing_enter()` *functions to wait. When the* `incoming_leave()` *or* `outgoing_leave()` *is called then a wake signal is sent to* `incoming_enter()` *and* `outgoing_enter()` *which allows the car to enter. Also the signal to* `incoming_enter()` *and* `outgoing_enter() is made when repairs are finished in street_thread().`

<u>Criteria #2:</u> We assume the street is too narrow for cars to be incoming and outgoing simultaneously. So, there will be either only incoming cars or only outgoing cars travelling at a time. However, the streaming should keep running without deadlock in either direction irrespective of how cars arrive.

Logic: *The logic that satisfy this criteria is that in the when* `outgoing_onstreet` *is greater than 0 then the* `incoming_enter()` *is put to wait and when* `incoming_onstreet` *is greater than 0 then the* `outgoing_enter()` *is put to wait.*

<u>Criteria #3:</u> After every 7th car leaves, the street becomes unusable and has to be repaired. Cars do not enter the street unless it is ready to use. Only the street thread is allowed to repair the street.

Logic: *The logic that satisfy this criteria is that in the when* `car_since_repair` *is less than 7 then the* `street_thread()` *is put to wait. The* `street_threat()` *also makes sure that when repairing there are no cars on the street. When the time for repair comes it also puts the* `incoming_enter()` *and* `outgoing_enter()` *to wait, so as the 7th car leave no new car enters until the street repairs are done. The wake signal to* `street_thread()` *is given when any car leaves the road.*