ECE 321: Software Requirements Engineering Assignment 3

Arun Woosaree

1514457

October 28, 2018

algebra QueueOfCars

imports Integer, Boolean;
introduces

sorts Queue, Car;

operations

New: → Queue;
CarArrives: Car x Queue → Queue;
CarDeparts: Queue → Queue;
IsEmpty: Queue → Boolean;
NumberOfCars: Queue → Integer;
Longer: Queue x Integer → Boolean;
FirstCar: Queue → Car;
Equal: Queue x Queue → Boolean;
WhichQueue: Queue x Queue x Car → Integer;
Position: Car x Queue → Integer;

constrains New, CarArrives, CarDeparts, IsEmpty, NumberOfCars, Longer, FirstCar, Equal, WhichQueue, Position, so that Queue generated by [New, CarArrives]

for all [q:Queue, c:Car, i:Integer]

CarDeparts(New) = error;
FirstCar(New) = error;
CarDeparts(CarArrives(q,c)) = q;
IsEmpty(New) = true;
IsEmpty(CarArrives(q,c)) = false;
Longer(CarArrives(q,c), i) =

    if (i > NumberOfCars(q)) then true;
    else false;

NumberOfCars(New) = 0;
NumberOfCars(CarArrives(q,c)) =

    NumberOfCars(CarDeparts(q)) + 1;

FirstCar(q) = c;
Equal(q1,q2) =

    if (NumberOfCars(q1) == NumberOfCars(q2)) then
    true;
    else false;

WhichQueue(q1,q2,c) =

    k what the fuck how do you do this

Position(c, q) =

    k what the fuck how do you do this