Assignment 1: Modelling Petrol Pumps

1 Petrol Pump Units

You are to build a controller of a petrol pump unit. The petrol pump unit should contain a number of independent pumps each can be conceptually modelled by an automata. The most common setup is one pump unit has 3 pumps with a variety of different possible fuels; 91, 95 and Diesel. Each pump draws from a separate reservoir, and may have a separate price. A person, in this case a main program, must not be able to crash the petrol pump unit.

To aid verification the actions of the **pumps must be defined using contracts**. If the customer had direct access to one of the pumps they could ignore the contract and crash the pump. To prevent this the customer must only access the pump unit. Only the pump unit has access to the individual pumps. You will have to write a pump unit that respects the contracts of the pumps and hence prevents system failure.

You need to demonstrate that the pump unit will not cheat the customer nor the petrol station. You may be able to achieve this by defining an invariant of the pump.

You must submit:

1. **SPARK Ada project**

2. **A brief description of the package structure you used**. This can include a brief justification if you think it helpful.

3. **A brief justification of code correctness**. This should out line techniques that you may have used:

   (a) Unit tests and assert statements

   (b) functional, flow or data contracts

   (c) coverage reports

4. Ambiguities in the specification may cause you to make some design decisions. List any that you think important. An empty list is perfectly acceptable.

2 Petrol Pump Automata

A petrol pump has various states and transitions that can be modelled as a finite state automata. We want to define a pump controller that is capable of calling device drivers and of responding to outside inputs. Define a boundary package with IO sub programs. The output sub programs could contain the device drivers and the input sub programs might be called by out side events. For this assignment, and for testing software, all IO sub programs need contain no more than print statements.

2.1 Pump Operation

A petrol pump has a nozzle, a reservoir and may be in one of several states.

- Base State: This is the normal state of the fuel pump. There are no outstanding fuel purchased and the nozzles are in their cradles.
- Ready State: When we lift the nozzle, we enter into a ready state. We can now start pumping fuel or return the nozzle to the cradle
- Pumping State: We have inserted the nozzle, and are pumping fuel. The pump has a sensor that determines if the car's fuel tank is full. We may not pump fuel if the tank is full. Similarly, we may not pump if the fuel reserve is empty.
- Waiting State: Once we have returned the nozzle to the cradle, we need to pay any outstanding amount for fuel consumed, or we can pick up the nozzle and continue pumping.

A customer can also take several actions that change the state of the pump.

- Lift Nozzle: When the nozzle is in the cradle (Base or Waiting), a customer may lift the nozzle to enter the Ready state.
- Replace Nozzle: When the nozzle is Ready, it may be returned to the cradle to enter the waiting state.
- Start Pumping: When the pump is ready, we can start pumping a predetermined amount of fuel to enter the pumping state.
- Stop Pumping: When pumping, we can the stop pumping to return to the ready state.

- Pay: Once we have returned the nozzle to the cradle, we must then pay any outstanding amount to return to the base state.

By the nature of the pumps the customer is unable to Start Pumping petrol unless the Nozzle has been picked up. The pump unit needs to enforce this, and other, restrictions.

Our petrol station has a single cash register. Payment for fuel must all be done at the same cash register for all pumps in the pump unit. Another restriction we make is that fuel may not be pumped from a pump until all bills at other pumps at the pump unit have been settled. Thus for example, if we pump diesel we must pay for it before we can pump unleaded.

2.2 Objectives

We would like to ensure the following properties of the pump unit.

- The pump unit must never fail. Only correct transitions are made. That is, a customer may never take an action unless the starting state allows it. As an example, a pump will not start pumping until the nozzle is lifted.

- Finally, we would like to ensure that at all steps we have balanced our books. That is, if fuel is consumed, then it must be paid for. This is a requirement across the entire pump unit. To help with this an invariant of each pump might be the sum of, the cash in the register, the fuel in the reservoir and the fuel that has yet to be paid for.