```
THEORY
  IMPORT THEORY PROJECTS
    /WaterTankTheory THEORIES /WaterTankTheory/Valves.dtf|org.eventb.theory.core.deployedTheoryRoot
      #Valves
  DATA TYPES
    TankState
    CONSTRUCTORS
      Normal()
      Emptying()
      Filling()
      Stable()
  OPERATORS
    isFlow <predicate> (state: TankState,DR: ℙ(RReal),Phi: RRealPlus ⇸ RReal,Qmin: RReal,Qmax:
      RReal)
      well-definedness DR ⊆ dom(Phi)
    isFlowODE <predicate> (state: TankState,DR: ℙ(RReal),Phip: RRealPlus×RReal ⇸ RReal,Qmin: RReal
      ,Qmax: RReal)
      well-definedness DR×Closed2Closed(Qmin,Qmax) ⊆ dom(Phip)
    isFlowEq <predicate> (state: TankState,DR: ℙ(RReal),eq: DE(RReal),Qmin: RReal,Qmax: RReal)
      well-definedness Solvable(DR,eq)
      direct definition
        ∀ Q · Q ∈ RRealPlus ⇸ RReal ∧ DR ⊆ dom(Q) ∧ solutionOf(DR,Q,eq) ⇒ isFlow(state,DR,Q,Qmin,
          Qmax)
    Delta <expression> (delta_in: RReal ,delta_out: RReal)
      well-definedness Rzero ↦ delta_in ∈ lt,Rzero ↦ delta_out ∈ lt
      direct definition
        (λ io_ · io_ ∈ InOutValve | minus(times(delta_in ↦ in_rstatus(io_)) ↦ times(delta_out ↦
          out_rstatus(io_))))
    flowIO <expression> (Qmin: RReal,Qmax: RReal,delta_in: RReal,delta_out: RReal)
      well-definedness Rzero ↦ Qmin ∈ leq,Rzero ↦ Qmax ∈ lt,Qmin ↦ Qmax ∈ leq,Rzero ↦ delta_in ∈
        lt,Rzero ↦ delta_out ∈ lt
      direct definition
        (λ io_ · io_ ∈ InOutValve |
          (λ t0_ ↦ q0_ · t0_ ∈ RRealPlus ∧ q0_ ∈ RReal |
            (λ t_ ↦ q_ ·
              t_ ∈ RRealPlus ∧ q_ ∈ RReal ∧
              plus(times(Delta(delta_in,delta_out)(io_) ↦ minus(t_ ↦ t0_)) ↦ q0_) ∈ Closed2Closed
                (Qmin,Qmax) |
                Delta(delta_in,delta_out)(io_)
            ) ∪
            (λ t_ ↦ q_ ·
              t_ ∈ RRealPlus ∧ q_ ∈ RReal ∧
              plus(times(Delta(delta_in,delta_out)(io_) ↦ minus(t_ ↦ t0_)) ↦ q0_) ∉ Closed2Closed
                (Qmin,Qmax) |
                Rzero
            )
          )
        )
    flowIOParts <expression> (Qmin: RReal,Qmax: RReal,delta_in: RReal,delta_out: RReal,iov:
      InOutValve,t0: RRealPlus,q0: RReal)
      well-definedness Rzero ↦ Qmin ∈ lt,Rzero ↦ Qmax ∈ lt,Rzero ↦ delta_in ∈ lt,Rzero ↦
        delta_out ∈ lt,Rzero ↦ q0 ∈ leq,q0 ↦ Qmax ∈ leq
      direct definition
        {
          { t_ | t_ ∈ RRealPlus ∧ plus(times(Delta(delta_in,delta_out)(iov) ↦ minus(t_ ↦ t0)) ↦ q0
            ) ∈ Closed2Closed(Qmin,Qmax) },
          { t_ | t_ ∈ RRealPlus ∧ plus(times(Delta(delta_in,delta_out)(iov) ↦ minus(t_ ↦ t0)) ↦ q0
            ) ↦ Qmin ∈ lt },
          { t_ | t_ ∈ RRealPlus ∧ Qmax ↦ plus(times(Delta(delta_in,delta_out)(iov) ↦ minus(t_ ↦ t0
            )) ↦ q0) ∈ lt }
        }
    FlowIOODE <expression> (Qmin: RReal,Qmax: RReal,delta_in: RReal,delta_out: RReal)
      well-definedness Rzero ↦ Qmin ∈ leq,Rzero ↦ Qmax ∈ lt,Qmin ↦ Qmax ∈ leq,Rzero ↦ delta_in ∈
        lt,Rzero ↦ delta_out ∈ lt
      direct definition
        (λ io_ · io_ ∈ InOutValve |
          (λ t0_ ↦ q0_ · t0_ ∈ RRealPlus ∧ q0_ ∈ RReal |
            ode(flowIO(Qmin,Qmax,delta_in,delta_out)(io_)(t0_↦q0_),q0_,t0_)
          )
        )
    SingleTankPolicy <expression> (mode: TankState)
    NoFlow <expression> ()
      direct definition
```

```
        (λ t_↦Q_ · t_ ∈ RRealPlus ∧ Q_ ∈ RReal | Rzero)
THEOREMS
  flowode_yields_flow:
    ∀ DR,Qmin,Qmax,Phip,st,t0,q0 ·
      DR ⊆ RRealPlus ∧
      Qmin ∈ RReal ∧ Rzero ↦ Qmin ∈ leq ∧
      Qmax ∈ RReal ∧ Rzero ↦ Qmax ∈ lt ∧
      Qmin ↦ Qmax ∈ leq ∧
      st ∈ TankState ∧
      t0 ∈ DR ∧ q0 ∈ RReal ∧
      Phip ∈ RRealPlus×RReal ⇸ RReal ∧
      DR×Closed2Closed(Qmin,Qmax) ⊆ dom(Phip) ∧
      isFlowODE(st,DR,Phip,Qmin,Qmax) ∧
      Solvable(DR,ode(Phip,q0,t0))
      ⇒ (
        ∀ Phi · Phi ∈ RRealPlus ⇸ RReal ∧ DR ⊆ dom(Phi) ∧ solutionOf(DR,Phi,ode(Phip,q0,t0)) ⇒
          isFlow(st,DR,Phi,Qmin,Qmax)
      )
  flow_type:
    ∀ Qmin,Qmax,delta_in,delta_out,iov,t0,q0 ·
      Qmin ∈ RReal ∧ Rzero ↦ Qmin ∈ leq ∧
      Qmax ∈ RReal ∧ Rzero ↦ Qmax ∈ lt ∧
      Qmin ↦ Qmax ∈ leq ∧
      delta_in ∈ RReal ∧ Rzero ↦ delta_in ∈ lt ∧
      delta_out ∈ RReal ∧ Rzero ↦ delta_out ∈ lt ∧
      iov ∈ InOutValve ∧
      t0 ∈ RRealPlus ∧
      q0 ∈ RReal ⇒ (
        flowIO(Qmin,Qmax,delta_in,delta_out)(iov)(t0↦q0) ∈ RRealPlus×RReal → RReal
      )
  flow_lipschitz:
    ∀ Qmin,Qmax,delta_in,delta_out,iov,t0,q0 ·
      Qmin ∈ RReal ∧ Rzero ↦ Qmin ∈ leq ∧
      Qmax ∈ RReal ∧ Rzero ↦ Qmax ∈ lt ∧
      Qmin ↦ Qmax ∈ leq ∧
      delta_in ∈ RReal ∧ Rzero ↦ delta_in ∈ lt ∧
      delta_out ∈ RReal ∧ Rzero ↦ delta_out ∈ lt ∧
      iov ∈ InOutValve ∧
      t0 ∈ RRealPlus ∧
      q0 ∈ RReal ∧ Rzero ↦ q0 ∈ leq ∧ q0 ↦ Qmax ∈ leq ⇒ (
        ∀ t_ · t_ ∈ RRealPlus ⇒
          lipschitzContinuous(RReal,RReal,partial2(flowIO(Qmin,Qmax,delta_in,delta_out)(iov)(t0↦
              q0),t_))
      )
  flow_piecewise_continuous:
    ∀ Qmin,Qmax,delta_in,delta_out,iov,t0,q0 ·
      Qmin ∈ RReal ∧ Rzero ↦ Qmin ∈ leq ∧
      Qmax ∈ RReal ∧ Rzero ↦ Qmax ∈ lt ∧
      Qmin ↦ Qmax ∈ leq ∧
      delta_in ∈ RReal ∧ Rzero ↦ delta_in ∈ lt ∧
      delta_out ∈ RReal ∧ Rzero ↦ delta_out ∈ lt ∧
      iov ∈ InOutValve ∧
      t0 ∈ RRealPlus ∧
      q0 ∈ RReal ∧ Rzero ↦ q0 ∈ leq ∧ q0 ↦ Qmax ∈ leq ⇒ (
        partialPiecewiseContinuous(
          flowIOParts(Qmax,Qmin,delta_in,delta_out,iov,t0,q0),
          RReal,RReal,
          flowIO(Qmin,Qmax,delta_in,delta_out)(iov)(t0↦q0)
        )
      )
  flow_pw_CL_cond:
    ∀ Qmin,Qmax,delta_in,delta_out,iov,t0,q0 ·
      Qmin ∈ RReal ∧ Rzero ↦ Qmin ∈ leq ∧
      Qmax ∈ RReal ∧ Rzero ↦ Qmax ∈ lt ∧
      Qmin ↦ Qmax ∈ leq ∧
      delta_in ∈ RReal ∧ Rzero ↦ delta_in ∈ lt ∧
      delta_out ∈ RReal ∧ Rzero ↦ delta_out ∈ lt ∧
      iov ∈ InOutValve ∧
      t0 ∈ RRealPlus ∧
      q0 ∈ RReal ∧ Rzero ↦ q0 ∈ leq ∧ q0 ↦ Qmax ∈ leq ⇒ (
        PiecewiseCauchyLipschitzCondition(
          flowIOParts(Qmin,Qmax,delta_in,delta_out,iov,t0,q0),
          RReal,
          FlowIOODE(Qmin,Qmax,delta_in,delta_out)(iov)(t0↦q0)
```

```
        )
      )
    flowio_is_flowode_single_tank_policy:
      ∀ DR, Qmin, Qmax, delta_in, delta_out, state ·
        DR ⊆ RRealPlus ∧
        Qmin ∈ RReal ∧ Rzero ↦ Qmin ∈ leq ∧
        Qmax ∈ RReal ∧ Rzero ↦ Qmax ∈ lt ∧
        Qmin ↦ Qmax ∈ leq ∧
        delta_in ∈ RReal ∧ Rzero ↦ delta_in ∈ lt ∧
        delta_out ∈ RReal ∧ Rzero ↦ delta_out ∈ lt ∧
        state ∈ TankState
        ⇒ (
          ∀ iov, t0, q0 ·
            iov ∈ SingleTankPolicy(state) ∧
            t0 ∈ DR ∧ q0 ∈ RReal ∧
            Qmin ↦ q0 ∈ leq ∧ q0 ↦ Qmax ∈ leq
            ⇒
              isFlowODE(state, DR, flowIO(Qmin, Qmax, delta_in, delta_out)(iov)(t0↦q0), Qmin, Qmax)
        )
    flow_LC:
      ∀ DR,
        Phi1, l, Q1min, Q1max,
        Phi2, m, Q2min, Q2max ·
        DR ⊆ RReal ∧
        l ∈ RReal ∧ Rzero ↦ l ∈ lt ∧
        Q1min ∈ RReal ∧ Q1max ∈ RReal ∧ Q1min ↦ Q1max ∈ leq ∧
        Rzero ↦ Q1min ∈ leq ∧ Rzero ↦ Q1max ∈ lt ∧
        Phi1 ∈ RRealPlus ⇸ RReal ∧ DR ⊆ dom(Phi1) ∧
        m ∈ RReal ∧ Rzero ↦ m ∈ lt ∧
        Q2min ∈ RReal ∧ Q2max ∈ RReal ∧ Q2min ↦ Q2max ∈ leq ∧
        Rzero ↦ Q2min ∈ leq ∧ Rzero ↦ Q2max ∈ lt ∧
        Phi2 ∈ RRealPlus ⇸ RReal ∧ DR ⊆ dom(Phi2) ⇒ (
          (∀ st · st ∈ TankState ⇒ isFlow(st, DR, Phi1, Q1min, Q1max) ∧ isFlow(st, DR, Phi2, Q2min, Q2max))
        ⇔
          (∀ st · st ∈ TankState ⇒ isFlow(
                        st,
                        DR,
                        LinComb2(l, Phi1, m, Phi2),
                        sLinComb2(l, Q1min, m, Q2min),
                        sLinComb2(l, Q1max, m, Q2max)
                        )
          )
        )
    flowODE_LC:
      ∀ DR,
        Phip1, l, Q1min, Q1max,
        Phip2, m, Q2min, Q2max ·
        DR ⊆ RReal ∧
        Q1min ∈ RReal ∧ Q1max ∈ RReal ∧ Q1min ↦ Q1max ∈ leq ∧
        Rzero ↦ Q1min ∈ leq ∧ Rzero ↦ Q1max ∈ lt ∧
        Phip1 ∈ RRealPlus×RReal ⇸ RReal ∧ DR×Closed2Closed(Q1min, Q1max) ⊆ dom(Phip1) ∧
        m ∈ RReal ∧ Rzero ↦ m ∈ lt ∧
        Q2min ∈ RReal ∧ Q2max ∈ RReal ∧ Q2min ↦ Q2max ∈ leq ∧
        Rzero ↦ Q2min ∈ leq ∧ Rzero ↦ Q2max ∈ lt ∧
        Phip2 ∈ RRealPlus×RReal ⇸ RReal ∧ DR×Closed2Closed(Q2min, Q2max) ⊆ dom(Phip2) ⇒ (
          (∀ st · st ∈ TankState ⇒ isFlowODE(st, DR, Phip1, Q1min, Q1max) ∧ isFlowODE(st, DR, Phip2, Q2min
              , Q2max))
        ⇔
          (∀ st · st ∈ TankState ⇒ isFlowODE(
                        st,
                        DR,
                        LinComb2(l, Phip1, m, Phip2),
                        sLinComb2(l, Q1min, m, Q2min),
                        sLinComb2(l, Q1max, m, Q2max)
                        )
          )
        )
    flowode_is_floweq:
      ∀ DR, Qmin, Qmax, Phip, st, t0, q0 ·
        DR ⊆ RRealPlus ∧
        Qmin ∈ RReal ∧ Rzero ↦ Qmin ∈ leq ∧
        Qmax ∈ RReal ∧ Rzero ↦ Qmax ∈ lt ∧
        Qmin ↦ Qmax ∈ leq ∧
        st ∈ TankState ∧
```

```
206             t0 ∈ DR ∧ q0 ∈ RReal ∧
207             Phip ∈ RRealPlus×RReal ↠ RReal ∧ DR×Closed2Closed(Qmin,Qmax) ⊆ dom(Phip) ∧
208             isFlowODE(st,DR,Phip,Qmin,Qmax) ∧
209             Solvable(DR,ode(Phip,q0,t0))
210             ⇒
211                isFlowEq(st,DR,ode(Phip,q0,t0),Qmin,Qmax)
212 END
```