```
uri "http://sadl.org/MinimalExampleEquations.sadl" alias mexeq.

import "http://sadl.org/ScientificConcepts1.sadl".

Equation velocityOfPhysicalObject(PhysicalObject o) returns PhysicalObject:
return o
        where v is a Velocity with ^value (derivative(position of o, ^time, 1)),
                with unit unitResolver("/", unit of position of o, unit of ^time of
position of o) and
                velocity of o is v.

Equation accelerationOfPhysicalObject1(PhysicalObject o) returns
PhysicalObject: return o
        where acc is an Acceleration with ^value av, with unit au and
                [av,au] is derivative(velocity of o, ^time, 1) and
                acceleration of o is acc.

Equation accelerationOfPhysicalObject2(PhysicalObject o) returns
PhysicalObject: return o
        where acc is an Acceleration with ^value (derivative(position of o,
^time, 2)),
                with unit unitResolver("/", unit of velocity of o, unit of ^time of
position of o) and
                acceleration of o is acc.

Equation momentumOfPhysicalObject(PhysicalObject o) returns
PhysicalObject: return o
         where v is velocity of velocityOfPhysicalObject(o) and
                p is a Momentum with ^value (^value of mass of o * ^value of v),
                with unit unitResolver("*", unit of mass of o, unit of v).

Equation newtons2ndLaw
        (note "net Force on a physical object is equal to the derivative of the
momentum of the object with respect to time.")
        (PhysicalObject o) returns Force: a Force f with ^value fv, with unit fu
                return f
                        where mv is a Momentum with ^value (^value of mass of o
* ^value of velocity of o),
                                with unit unitResolver("*", unit of mass of o, unit of
velocity of o) and
                                [fv,fu] = derivative(mv, ^time, 1).
Equation newtons2ndLawConstantMass
        (note "net Force on a physical object is equal to the mass of the object times
its acceleration for constant mass.")
        (PhysicalObject o) returns Force: a Force f with ^value fv, with unit fu
                return f
                        where acc is an Acceleration with ^value accv, with unit
accu and
                                [mdotv, mdotu] is derivative(mass of o, ^time, 1)
and mdotv is 0 and
                                [accv,accu] = derivative(velocity of o, ^time, 1) and
                                fv = ^value of mass of o * accv and
                                fu = unitResolver("*", unit of mass of o, accu).

 External mass-acceleration (Mass m, acceleration acc) returns Force:
```

"http://com.ge.research.darpa.answer/probabilisticNewtonsSecondLaw".

dependsOn describes **ScientificConcept** with values of type **ScientificConcept**.
// from Equations above,
Velocity dependsOn Position.
Acceleration dependsOn Velocity.
Momentum dependsOn Mass, **dependsOn Velocity**.
Force dependsOn Mass, **dependsOn Velocity**, **dependsOn Acceleration**.