

This file demonstrates features of MathSBML

Input is emphasized with a yellow background

The Repressilator

Based on the model of:
Elowitz, M.B., and Leibler, S. (2000) A synthetic oscillatory network of transcriptional regulators. *Nature*, 403:335-338,

```
<< mathsbml.m
```

MathSBML 2.2.1-beta-17 (25-Feb-2004) loaded 27-February-2004 18:19:14.648695

read in an SBML file and display the verbose listing

```
m = SBMLRead["repressilator.xml", verbose -> True];
```

File Name:repressilator.xml
SBML Level 2 Version 1

Model: Repressilator Model = repressilator

Function Definitions

----- None -----

Unit Definitions

----- None -----

Compartments

<u>ID</u>	<u>Name</u>	<u>Dimension</u>	<u>Size</u>	<u>Units</u>	<u>Derived Units</u>	<u>Outside</u>
cell	cell	3	...	volume	volume	...

Species

<u>ID</u>	<u>Name</u>	<u>Compartment</u>	<u>Type</u>	<u>I.C.</u>	<u>Units</u>	<u>Derived</u>
x	x	cell
y	y	cell
z	z	cell
py	py	cell
px	px	cell	amount	5.	substance	substance
pz	pz	cell	amount	15.	substance	substance
null	null	cell

Global Parameters

<u>ID</u>	<u>Name</u>	<u>Value</u>	<u>Units</u>	<u>Derived Units</u>	<u>Constant</u>
alpha	alpha	250.	True
beta	beta	5.	True
alpha0	alpha0	0	True
alpha1	alpha1	0	True
n	n	2.1	True
k1	k1	1.	True
K	K	1.	True

Rules

----- None -----

Reactions, global and local contexts suppressed

<u>ID</u>	<u>Name</u>	<u>Fast</u>	<u>Reaction</u>	<u>Modifiers</u>	<u>Parameters</u>
reaction1	reaction1	False	null \rightleftharpoons x	pz	...
reaction2	reaction2	False	null \rightleftharpoons y	px	...
reaction3	reaction3	False	null \rightleftharpoons z	py	...
reaction4	reaction4	False	px \rightleftharpoons null	x	...
reaction5	reaction5	False	py \rightleftharpoons null	y	...
reaction6	reaction6	False	pz \rightleftharpoons null	z	...

Differential Equations, global context suppressed

Variable

px

py

pz

x

y

z

ODEs $px'[t] == 5 * (-px[t] + x[t])$ $py'[t] == 5 * (-py[t] + y[t])$ $pz'[t] == 5 * (-pz[t] + z[t])$ $x'[t] == 250 / (1 + pz[t]^2.1) - 1 * x[t]$ $y'[t] == 250 / (1 + px[t]^2.1) - 1 * y[t]$ $z'[t] == 250 / (1 + py[t]^2.1) - 1 * z[t]$ **Events**

----- None -----

read file, convert to XPP**■ write to a file**

```
SBMLWrite[inputfile -> "repressilator.xml", outputfile -> "rep.ode", format -> "XPP"]
```

rep.ode

■ same thing, but write output to the screen

```
SBMLWrite[inputfile -> "repressilator.xml", format -> "XPP"]
```

```
# Model Name:      Repressilator Model
#
# Creation Time: 26-November-2003 10:39:46.983614
# User:           jamestkirk
# Machine:        DHCP-52-160
# System:         PowerPC PowerMac MacOSX
# Generated by MathSBML 2.2.1-beta-9 (25 Nov 2003)
#
# Differential Equations
#
repressilator_px'=5.*(-repressilator_px+repressilator_x)
repressilator_py'=5.*(-repressilator_py+repressilator_y)
repressilator_pz'=5.*(-repressilator_pz+repressilator_z)
repressilator_x'=250./(1.+repressilator_pz^2.1)-1.*repressilator_x
repressilator_y'=250./(1.+repressilator_px^2.1)-1.*repressilator_y
repressilator_z'=250./(1.+repressilator_py^2.1)-1.*repressilator_z
#
#
# Parameters
#
par repressilator_alpha=250.
par repressilator_beta=5.
par repressilator_alpha0=0
par repressilator_alpha1=0
par repressilator_n=2.1
par repressilator_k1=1.
par repressilator_K=1.
#
#
# Initial Conditions
#
init repressilator_x=Indeterminate
init repressilator_y=Indeterminate
init repressilator_z=Indeterminate
init repressilator_py=Indeterminate
init repressilator_px=5.
init repressilator_pz=15.
#
done
```

read file, convert to Fortran

■ write to a file

```
SBMLWrite[inputfile -> "repressilator.xml",
  outputfile -> "rep.for", format -> "FORTRAN"]
```

```
rep.for
```

■ same thing, but write output to the screen

```
SBMLWrite[inputfile -> "repressilator.xml", format -> "FORTRAN"]
```

```
C SBML Model Name:  Repressilator Model
C
C Generated by MathSBML 2.2.1-beta-9 (25 Nov 2003)
C Creation Time: 26-November-2003 10:40:58.746876
C User:           jamestkirk
C Machine:        DHCP-52-160
C Processor:      PowerPC
C Machine type:   PowerMac
C Oper. System:   MacOSX
C -----
C
C lsodi compliant SBML model
C Reference: http://netlib.org/alliant/ode/prog/lsodi.f
C
C -----
C
C This file contains the following modules:
C
C Module Name      Description
C -----
C addp (subroutine) Add A to any matrix - required by lsodi
C init (subroutine) Set initial conditions
C jac (subroutine)  Compute Jacobian - required by lsodi
C res (subroutine)  Calculate Residuals - required by lsodi
C
C -----
C
C      subroutine res(neq,t,y,s,r,ires)
C      double precision r,s,t,y
C
C This is subroutine res for lsodi
C This function computes the residuals  $r(i)=g(t,y)-A(t,y)(dy/dt)$ 
C for the linear-implicit system  $(A)*(dy/dt)=g(t,y)$ 
C where A is a constant, possibly singular, matrix.
C
C Here A is diagonal with (restriction imposed by SBML, not lsodi)
```

```

C      A(i,i)=1, i=1,...,m, where m=# of odes in the SBML Model
C      A(i,i)=0, i=m+1,...,m+nrules, where nrules =# of algebraic rules
C      A(i,i)=0, i=nrules+1,...,nvars, where nvars is the total number
C          of variables in the system and nvars-nrules-m>0 is the
C          number of variables controlled purely by events
C
C If the system is purely differential
C the right-hand side of the system dydt = g(y,t) will
C be returned if s is zero-filled.
C
C Representation of Model Variables by the array y
C -----
C y(1) = px
C y(2) = py
C y(3) = pz
C y(4) = x
C y(5) = y
C y(6) = z
C
      dimension r(6),s(6),y(6)
      r(1)=-s(1) + 5.*(-y(1) + y(4))
      r(2)=-s(2) + 5.*(-y(2) + y(5))
      r(3)=-s(3) + 5.*(-y(3) + y(6))
      r(4)=-s(4) + 250./(1. + y(3)**2.1) - 1.*y(4)
      r(5)=-s(5) + 250./(1. + y(1)**2.1) - 1.*y(5)
      r(6)=-s(6) + 250./(1. + y(2)**2.1) - 1.*y(6)
      return
      end
C
C
C -----
C
      subroutine addp(neq, t, y, ml, mu, p, nrowp)
C
C Subroutine addp required by lsodi
C
      double precision p, t, y
      dimension y(6), p(nrowp,6)
      integer i
      Do i = 1,6
          p(i,i) = p(i,i)+1
      End Do
      return
      end
C
C -----
C
      subroutine jac (neq,t,y,s,ml,mu,p,nrowp)
C
C Subroutine jac required by lsodi, computes jacobian
C
      dimension y(6), s(6),p(nrowp,6)

      p(1,1)=-5.
      p(1,2)=0
      p(1,3)=0
      p(1,4)=5.
      p(1,5)=0

```

```
p(1,6)=0
p(2,1)=0
p(2,2)=-5.
p(2,3)=0
p(2,4)=0
p(2,5)=5.
p(2,6)=0
p(3,1)=0
p(3,2)=0
p(3,3)=-5.
p(3,4)=0
p(3,5)=0
p(3,6)=5.
p(4,1)=0
p(4,2)=0
p(4,3)=(-525.*y(3)**1.1)/(1.+y(3)**2.1)**2
p(4,4)=-1.
p(4,5)=0
p(4,6)=0
p(5,1)=(-525.*y(1)**1.1)/(1.+y(1)**2.1)**2
p(5,2)=0
p(5,3)=0
p(5,4)=0
p(5,5)=-1.
p(5,6)=0
p(6,1)=0
p(6,2)=(-525.*y(2)**1.1)/(1.+y(2)**2.1)**2
p(6,3)=0
p(6,4)=0
p(6,5)=0
p(6,6)=-1.

return
end

C
C -----
C
Subroutine init(neq,y)
double precision y
dimension y(6)
y(1)=5.
y(2)=Indeterminate
y(3)=15.
y(4)=Indeterminate
y(5)=Indeterminate
y(6)=Indeterminate
return
end
```

read file, display as web page

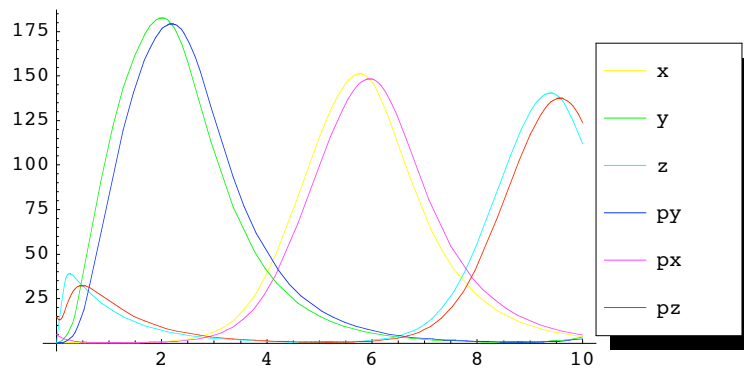
```
SBMLWrite[inputfile -> "repressilator.xml", outputfile -> "rep.htm", format -> "HTML"]
```

rep.htm

read file and perform simulation

```
m = SBMLRead["repressilator.xml"];  
s = SBMLNDSolve[m, 10];  
SBMLPlot[s];
```

>>Warning: The following variables have indeterminate initial conditions that have been assumed to be zero: x,y,z,py



Create SBML model for repressilator

```
newModel["repressilator", name → "Repressilator Model", echo → False];
addCompartment[cell];
addSpecies[{x, y, z, py}];
addSpecies[px, initialAmount → 5];
addSpecies[pz, initialAmount → 15];
addParameter[alpha, value → 250];
addParameter[beta, value → 5];
addParameter[alpha0, value → 0];
addParameter[alpha1, value → 0];
addParameter[n, value → 2.1];
addParameter[k1, value → 1];
addParameter[K, value → 1];
addSpecies[null, boundaryCondition → True, constant → True];
addReaction[null → x, modifiers → pz,
  kineticLaw → -k1 * x + alpha0 + (alpha + alpha1 * pz^n) / (K^n + pz^n)];
addReaction[null → y, modifiers → px,
  kineticLaw → -k1 * y + alpha0 + (alpha + alpha1 * px^n) / (K^n + px^n)];
addReaction[null → z, modifiers → py,
  kineticLaw → -k1 * z + alpha0 + (alpha + alpha1 * py^n) / (K^n + py^n)];
addReaction[px → null, modifiers → x, kineticLaw → -beta * (x - px)];
addReaction[py → null, modifiers → y, kineticLaw → -beta * (y - py)];
addReaction[pz → null, modifiers → z, kineticLaw → -beta * (z - pz)];
createModel["repressilator.xml"]
```

repressilator.xml

Load SBML file and manipulate using the model builder API

■ load the file

```
loadModelBuilder["repressilator.xml"];

Compartment cell added.

Species x added to compartment cell
Species y added to compartment cell
Species z added to compartment cell
Species py added to compartment cell
Species px added to compartment cell
Species pz added to compartment cell
Species null added to compartment cell

Parameter alpha added.
Parameter beta added.
Parameter alpha0 added.
Parameter alpha1 added.
Parameter n added.
Parameter k1 added.
Parameter K added.

Reaction reaction1 added.
Reaction reaction2 added.
Reaction reaction3 added.
Reaction reaction4 added.
Reaction reaction5 added.
Reaction reaction6 added.

SBML Model repressilator loaded into Model Builder
```

■ display the current model

```
showModel[];
```

File Name:Internal Model SBML Level 2 Version 1

Model: Repressilator Model = repressilator

Function Definitions

----- None -----

Unit Definitions

----- None -----

Compartments

<u>ID</u>	<u>Name</u>	<u>Dimension</u>	<u>Size</u>	<u>Units</u>	<u>Derived Units</u>	<u>Outside</u>
cell	cell	3	...	volume	volume	...

Species

<u>ID</u>	<u>Name</u>	<u>Compartment</u>	<u>Type</u>	<u>I.C.</u>	<u>Units</u>	<u>Derived</u>
x	x	cell
y	y	cell
z	z	cell
py	py	cell
px	px	cell	amount	5.	substance	substance
pz	pz	cell	amount	15.	substance	substance
null	null	cell

Global Parameters

<u>ID</u>	<u>Name</u>	<u>Value</u>	<u>Units</u>	<u>Derived Units</u>	<u>Constant</u>
alpha	alpha	250.	True
beta	beta	5.	True
alpha0	alpha0	0	True
alpha1	alpha1	0	True
n	n	2.1	True
k1	k1	1.	True
K	K	1.	True

Rules

----- None -----

Reactions, global and local contexts suppressed

<u>ID</u>	<u>Name</u>	<u>Fast</u>	<u>Reaction</u>	<u>Modifiers</u>	<u>Parameters</u>
reaction1	reaction1	False	null ⇒ x	pz	...
reaction2	reaction2	False	null ⇒ y	px	...
reaction3	reaction3	False	null ⇒ z	py	...
reaction4	reaction4	False	px ⇒ null	x	...
reaction5	reaction5	False	py ⇒ null	y	...
reaction6	reaction6	False	pz ⇒ null	z	...

Differential Equations, global context suppressed

<u>Variable</u>	<u>ODEs</u>
px	$px'[t] == \text{beta} * (-px[t] + x[t])$
py	$py'[t] == \text{beta} * (-py[t] + y[t])$
pz	$pz'[t] == \text{beta} * (-pz[t] + z[t])$
x	$x'[t] == \text{alpha0} + (\text{alpha} + \text{alpha1} * pz[t]^n) / (K^n + pz[t]^n) - k1 * x[t]$
y	$y'[t] == \text{alpha0} + (\text{alpha} + \text{alpha1} * px[t]^n) / (K^n + px[t]^n) - k1 * y[t]$
z	$z'[t] == \text{alpha0} + (\text{alpha} + \text{alpha1} * py[t]^n) / (K^n + py[t]^n) - k1 * z[t]$

Events

----- None -----

- add a reaction
- observe that the species A and B have not been previously defined so they are first added to the model in the single compartment cell

```
addReaction[A → B, kineticLaw → k5 * A * x / (k6 + x), parameters → {k5 → 0.1, k6 → 2}]
```

```
Species A added to compartment cell
```

```
Species B added to compartment cell
```

```
Reaction reaction7 added.
```

```
showModel[];
```

File Name:Internal Model SBML Level 2 Version 1

Model: Repressilator Model = repressilator

Function Definitions

----- None -----

Unit Definitions

----- None -----

Compartments

<u>ID</u>	<u>Name</u>	<u>Dimension</u>	<u>Size</u>	<u>Units</u>	<u>Derived Units</u>	<u>Outside</u>
cell	cell	3	...	volume	volume	...

Species

<u>ID</u>	<u>Name</u>	<u>Compartment</u>	<u>Type</u>	<u>I.C.</u>	<u>Units</u>	<u>Derived</u>
x	x	cell
y	y	cell
z	z	cell
py	py	cell
px	px	cell	amount	5.	substance	substance
pz	pz	cell	amount	15.	substance	substance
null	null	cell
A	A	cell
B	B	cell

Global Parameters

<u>ID</u>	<u>Name</u>	<u>Value</u>	<u>Units</u>	<u>Derived Units</u>	<u>Constant</u>
alpha	alpha	250.	True
beta	beta	5.	True
alpha0	alpha0	0	True
alpha1	alpha1	0	True
n	n	2.1	True
k1	k1	1.	True
K	K	1.	True

Rules

----- None -----

Reactions, global and local contexts suppressed

<u>ID</u>	<u>Name</u>	<u>Fast</u>	<u>Reaction</u>	<u>Modifiers</u>	<u>Parameters</u>
reaction1	reaction1	False	null \rightleftharpoons x	pz	...
reaction2	reaction2	False	null \rightleftharpoons y	px	...
reaction3	reaction3	False	null \rightleftharpoons z	py	...
reaction4	reaction4	False	px \rightleftharpoons null	x	...
reaction5	reaction5	False	py \rightleftharpoons null	y	...
reaction6	reaction6	False	pz \rightleftharpoons null	z	...
reaction7	reaction7	False	A \rightleftharpoons B	...	k5 \rightarrow 0.1 k6 \rightarrow 2.

Differential Equations, global context suppressed

<u>Variable</u>	<u>ODEs</u>
A	$A'[t] == -((\text{reaction7}^k5 * A[t] * x[t]) / (\text{reaction7}^k6 + x[t]))$
B	$B'[t] == (\text{reaction7}^k5 * A[t] * x[t]) / (\text{reaction7}^k6 + x[t])$
px	$px'[t] == \text{beta} * (-px[t] + x[t])$
py	$py'[t] == \text{beta} * (-py[t] + y[t])$
pz	$pz'[t] == \text{beta} * (-pz[t] + z[t])$
x	$x'[t] == \text{alpha0} + (\text{alpha} + \text{alpha1} * pz[t]^n) / (K^n + pz[t]^n) - k1 * x[t]$
y	$y'[t] == \text{alpha0} + (\text{alpha} + \text{alpha1} * px[t]^n) / (K^n + px[t]^n) - k1 * y[t]$
z	$z'[t] == \text{alpha0} + (\text{alpha} + \text{alpha1} * py[t]^n) / (K^n + py[t]^n) - k1 * z[t]$

Events

----- None -----

- realize that you forgot to set initial conditions on A and B, so set them now

```
modifySpecies[A, initialAmount  $\rightarrow$  5];
modifySpecies[B, initialAmount  $\rightarrow$  0];
```

Previous occurrence of species A removed.

Species A added to compartment cell

Previous occurrence of species B removed.

Species B added to compartment cell

- Show the model again

```
showModel[];
```

File Name:Internal Model
SBML Level 2 Version 1

Model: Repressilator Model = repressilator

Function Definitions

----- None -----

Unit Definitions

----- None -----

Compartments

<u>ID</u>	<u>Name</u>	<u>Dimension</u>	<u>Size</u>	<u>Units</u>	<u>Derived Units</u>	<u>Outside</u>
cell	cell	3	...	volume	volume	...

Species

<u>ID</u>	<u>Name</u>	<u>Compartment</u>	<u>Type</u>	<u>I.C.</u>	<u>Units</u>	<u>Derived</u>
x	x	cell
y	y	cell
z	z	cell
py	py	cell
px	px	cell	amount	5.	substance	substance
pz	pz	cell	amount	15.	substance	substance
null	null	cell
A	A	cell	amount	5.	substance	substance
B	B	cell	amount	0	substance	substance

Global Parameters

<u>ID</u>	<u>Name</u>	<u>Value</u>	<u>Units</u>	<u>Derived Units</u>	<u>Constant</u>
alpha	alpha	250.	True
beta	beta	5.	True
alpha0	alpha0	0	True
alpha1	alpha1	0	True
n	n	2.1	True
k1	k1	1.	True
K	K	1.	True

Rules

----- None -----

Reactions, global and local contexts suppressed

<u>ID</u>	<u>Name</u>	<u>Fast</u>	<u>Reaction</u>	<u>Modifiers</u>	<u>Parameters</u>
reaction1	reaction1	False	null ⇒ x	pz	...
reaction2	reaction2	False	null ⇒ y	px	...
reaction3	reaction3	False	null ⇒ z	py	...
reaction4	reaction4	False	px ⇒ null	x	...
reaction5	reaction5	False	py ⇒ null	y	...
reaction6	reaction6	False	pz ⇒ null	z	...
reaction7	reaction7	False	A ⇒ B	...	k5→0.1 k6→2.

Differential Equations, global context suppressed

<u>Variable</u>	<u>ODEs</u>
A	$A'[t] == -((\text{reaction7}^k5 * A[t] * x[t]) / (\text{reaction7}^k6 + x[t]))$
B	$B'[t] == (\text{reaction7}^k5 * A[t] * x[t]) / (\text{reaction7}^k6 + x[t])$
px	$px'[t] == \text{beta} * (-px[t] + x[t])$
py	$py'[t] == \text{beta} * (-py[t] + y[t])$
pz	$pz'[t] == \text{beta} * (-pz[t] + z[t])$
x	$x'[t] == \text{alpha0} + (\text{alpha} + \text{alpha1} * pz[t]^n) / (K^n + pz[t]^n) - k1 * x[t]$
y	$y'[t] == \text{alpha0} + (\text{alpha} + \text{alpha1} * px[t]^n) / (K^n + px[t]^n) - k1 * y[t]$
z	$z'[t] == \text{alpha0} + (\text{alpha} + \text{alpha1} * py[t]^n) / (K^n + py[t]^n) - k1 * z[t]$

Events

----- None -----

- Everything looks good so create the sbml and write it to the screen

```
createModel[]
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Generated 26-November-2003 11:32:14.887787 -->
<!-- Generated by MathSBML 2.2.1-beta-9 (25 Nov 2003) -->
<sbml xmlns="http://www.sbml.org/sbml/level2"
  level="2"
  version="1">
<model id="repressilator"
  name="Repressilator Model">
  <listOfFunctionDefinitions/>
  <listOfUnitDefinitions/>
  <listOfCompartments>
    <compartment id="cell"
      name="cell"/>
  </listOfCompartments>
  <listOfSpecies>
    <species id="x"
      name="x"
      compartment="cell"
      boundaryCondition="false"
      constant="false"/>
    <species id="y"
      name="y"
      compartment="cell"
      boundaryCondition="false"
      constant="false"/>
    <species id="z"
      name="z"
      compartment="cell"
      boundaryCondition="false"
      constant="false"/>
    <species id="py"
      name="py"
      compartment="cell"
      boundaryCondition="false"
      constant="false"/>
    <species id="px"
      name="px"
      compartment="cell"
      boundaryCondition="false"
      constant="false"
      initialAmount="5"/>
    <species id="pz"
      name="pz"
      compartment="cell"
      boundaryCondition="false"
      constant="false"
      initialAmount="15"/>
    <species id="null"
      name="null"
      compartment="cell"
      boundaryCondition="true"
      constant="true"/>
    <species id="A"
      name="A"
      compartment="cell"
      boundaryCondition="false"
      constant="false"
```

```
    initialAmount="5." />
<species id="B"
  name="B"
  compartment="cell"
  boundaryCondition="false"
  constant="false"
  initialAmount="0" />
</listOfSpecies>
<listOfParameters>
  <parameter id="alpha"
    name="alpha"
    value="250" />
  <parameter id="beta"
    name="beta"
    value="5" />
  <parameter id="alpha0"
    name="alpha0"
    value="0" />
  <parameter id="alpha1"
    name="alpha1"
    value="0" />
  <parameter id="n"
    name="n"
    value="2.1" />
  <parameter id="k1"
    name="k1"
    value="1" />
  <parameter id="K"
    name="K"
    value="1" />
</listOfParameters>
<listOfRules />
<listOfReactions>
  <reaction id="reaction1"
    name="reaction1"
    reversible="true"
    fast="false">
    <listOfReactants>
      <speciesReference species="null" />
    </listOfReactants>
    <listOfProducts>
      <speciesReference species="x" />
    </listOfProducts>
    <listOfModifiers>
      <modifierSpeciesReference species="pz" />
    </listOfModifiers>
  <kineticLaw timeUnits="time"
    substanceUnits="substance">
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <apply>
        <plus />
        <ci>alpha0</ci>
        <apply>
          <times />
          <apply>
            <plus />
            <apply>
              <times />
```

```
<ci>alpha</ci>
<apply>
  <power/>
  <ci>pz</ci>
  <ci>n</ci>
</apply>
</apply>
<ci>alpha</ci>
</apply>
<apply>
  <power/>
  <apply>
    <plus/>
    <apply>
      <power/>
      <ci>K</ci>
      <ci>n</ci>
    </apply>
    <apply>
      <power/>
      <ci>pz</ci>
      <ci>n</ci>
    </apply>
  </apply>
  <cn type="integer">-1</cn>
</apply>
</apply>
<apply>
  <times/>
  <cn type="integer">-1</cn>
  <apply>
    <times/>
    <ci>k1</ci>
    <ci>x</ci>
  </apply>
</apply>
</math>
</listOfParameters/>
</kineticLaw>
</reaction>
<reaction id="reaction2"
  name="reaction2"
  reversible="true"
  fast="false">
  <listOfReactants>
    <speciesReference species="null"/>
  </listOfReactants>
  <listOfProducts>
    <speciesReference species="y"/>
  </listOfProducts>
  <listOfModifiers>
    <modifierSpeciesReference species="px"/>
  </listOfModifiers>
  <kineticLaw timeUnits="time"
    substanceUnits="substance">
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <apply>
```

```

<plus/>
<ci>alpha0</ci>
<apply>
  <times/>
  <apply>
    <plus/>
    <apply>
      <times/>
      <ci>alpha1</ci>
      <apply>
        <power/>
        <ci>px</ci>
        <ci>n</ci>
      </apply>
    </apply>
  </apply>
  <ci>alpha</ci>
</apply>
<apply>
  <power/>
  <apply>
    <plus/>
    <apply>
      <power/>
      <ci>K</ci>
      <ci>n</ci>
    </apply>
    <apply>
      <power/>
      <ci>px</ci>
      <ci>n</ci>
    </apply>
  </apply>
  <cn type="integer">-1</cn>
</apply>
</apply>
<apply>
  <times/>
  <cn type="integer">-1</cn>
  <apply>
    <times/>
    <ci>k1</ci>
    <ci>y</ci>
  </apply>
</apply>
</math>
<listOfParameters/>
</kineticLaw>
</reaction>
<reaction id="reaction3"
  name="reaction3"
  reversible="true"
  fast="false">
<listOfReactants>
  <speciesReference species="null"/>
</listOfReactants>
<listOfProducts>
  <speciesReference species="z"/>

```

```
</listOfProducts>
<listOfModifiers>
  <modifierSpeciesReference species="py"/>
</listOfModifiers>
<kineticLaw timeUnits="time"
  substanceUnits="substance">
  <math xmlns="http://www.w3.org/1998/Math/MathML">
    <apply>
      <plus/>
      <ci>alpha0</ci>
      <apply>
        <times/>
        <apply>
          <plus/>
          <apply>
            <times/>
            <ci>alpha1</ci>
            <apply>
              <power/>
              <ci>py</ci>
              <ci>n</ci>
            </apply>
          </apply>
        </apply>
        <ci>alpha</ci>
      </apply>
      <apply>
        <power/>
        <apply>
          <plus/>
          <apply>
            <power/>
            <ci>K</ci>
            <ci>n</ci>
          </apply>
          <apply>
            <power/>
            <ci>py</ci>
            <ci>n</ci>
          </apply>
        </apply>
        <cn type="integer">-1</cn>
      </apply>
    </apply>
  </math>
  <listOfParameters/>
</kineticLaw>
</reaction>
<reaction id="reaction4"
```

```
    name="reaction4"
    reversible="true"
    fast="false">
<listOfReactants>
  <speciesReference species="px"/>
</listOfReactants>
<listOfProducts>
  <speciesReference species="null"/>
</listOfProducts>
<listOfModifiers>
  <modifierSpeciesReference species="x"/>
</listOfModifiers>
<kineticLaw timeUnits="time"
  substanceUnits="substance">
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <apply>
    <times/>
    <apply>
      <times/>
      <cn type="integer">-1</cn>
      <ci>beta</ci>
    </apply>
    <apply>
      <plus/>
      <ci>x</ci>
      <apply>
        <times/>
        <cn type="integer">-1</cn>
        <ci>px</ci>
      </apply>
    </apply>
  </math>
</listOfParameters/>
</kineticLaw>
</reaction>
<reaction id="reaction5"
  name="reaction5"
  reversible="true"
  fast="false">
<listOfReactants>
  <speciesReference species="py"/>
</listOfReactants>
<listOfProducts>
  <speciesReference species="null"/>
</listOfProducts>
<listOfModifiers>
  <modifierSpeciesReference species="y"/>
</listOfModifiers>
<kineticLaw timeUnits="time"
  substanceUnits="substance">
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <apply>
    <times/>
    <apply>
      <times/>
      <cn type="integer">-1</cn>
      <ci>beta</ci>
```

```

    </apply>
  <apply>
    <plus/>
    <ci>y</ci>
    <apply>
      <times/>
      <cn type="integer">-1</cn>
      <ci>py</ci>
    </apply>
  </apply>
</math>
</listOfParameters/>
</kineticLaw>
</reaction>
<reaction id="reaction6"
  name="reaction6"
  reversible="true"
  fast="false">
  <listOfReactants>
    <speciesReference species="pz"/>
  </listOfReactants>
  <listOfProducts>
    <speciesReference species="null"/>
  </listOfProducts>
  <listOfModifiers>
    <modifierSpeciesReference species="z"/>
  </listOfModifiers>
  <kineticLaw timeUnits="time"
    substanceUnits="substance">
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <apply>
        <times/>
        <apply>
          <times/>
          <cn type="integer">-1</cn>
          <ci>beta</ci>
        </apply>
        <apply>
          <plus/>
          <ci>z</ci>
          <apply>
            <times/>
            <cn type="integer">-1</cn>
            <ci>pz</ci>
          </apply>
        </apply>
      </math>
    </listOfParameters/>
  </kineticLaw>
</reaction>
<reaction id="reaction7"
  name="reaction7"
  reversible="true"
  fast="false">
  <listOfReactants>
    <speciesReference species="A"/>

```

```
</listOfReactants>
<listOfProducts>
  <speciesReference species="B"/>
</listOfProducts>
<listOfModifiers/>
<kineticLaw timeUnits="time"
  substanceUnits="substance">
  <math xmlns="http://www.w3.org/1998/Math/MathML">
    <apply>
      <times/>
      <ci>A</ci>
      <ci>k5</ci>
      <ci>x</ci>
      <apply>
        <power/>
        <apply>
          <plus/>
          <ci>k6</ci>
          <ci>x</ci>
        </apply>
        <cn type="integer">-1</cn>
      </apply>
    </apply>
  </math>
  <listOfParameters>
    <parameter id="k5" value="0.1">notesannotations</parameter>
    <parameter id="k6" value="2">notesannotations</parameter>
  </listOfParameters>
</kineticLaw>
</reaction>
</listOfReactions>
<listOfEvents/>
</model>
</sbml>
```