

### Uitwerking Huiswerkgave Inleiding Modelleren – Hoofdstuk 3

1a. We werken het geval voor het tandenpoetsen uit. De concepten zijn (we gebruiken Engelse termen en afkortingen):

```
tube=[cap:{open,close},hand:{l,r,none}]  
brush=[wh:{glass,mouth,else},paste:{yes,no},hand:{l,r,none}]
```

1b (i). Voor de tube zijn zes toestanden (cap:2 X hand:3). Alle toestandsovergangen waarbij precies één eigenschap verandert zijn toegestaan mits voldaan is aan de volgende eisen (we schrijven 'transitie:conditie//commentaar' en daarmee bedoelen we dat de transitie alleen mag als aan de conditie voldaan is; '!=' betekent 'is niet gelijk aan'; '^' betekent 'en'; 'v' betekent 'of'; '\*' betekent: 'elke waarde'):

veranderingen van tube.cap:

```
tube.cap==open → tube.cap==close : tube.hand !=none  
// je kunt het dopje niet dicht doen tenzij je de tube in je hand hebt  
tube.cap==close → tube.cap==open : tube.hand !=none  
// je kunt het dopje niet open doen tenzij je de tube in je hand hebt
```

veranderingen van tube.hand:

```
tube.hand==* → tube.hand==none : tube.cap==close  
// je legt de tube niet neer zonder dat er een dopje op zit
```

Voor de borstel zijn 18 toestanden (wh:3 X paste:2 X hand:3). Alle toestandsovergangen waarbij precies één eigenschap verandert zijn toegestaan mits voldaan is aan de volgende eisen:

veranderingen van brush.wh:

```
brush.wh==glass → brush.wh==mouth : brush.hand !=none ^ brush.paste==yes  
// stop alleen een tandenborstel in je mond als er tandpasta op zit  
brush.wh==glass → brush.wh==else : brush.hand !=none  
brush.wh==mouth → brush.wh==glass : brush.hand !=none ^ brush.paste==no  
// zet geen tandenborstel met tandpasta erop in een glas  
brush.wh==mouth → brush.wh==else : brush.hand !=none  
brush.wh==else → brush.wh==mouth : brush.hand !=none ^ brush.paste==yes  
// stop alleen een tandenborstel in je mond als er tandpasta op zit  
brush.wh==else → brush.wh==glass : brush.hand !=none ^ brush.paste==no  
// zet geen tandenborstel met tandpasta erop in een glas  
// voor bovenstaande 6 overgangen geldt voorts:  
// de tandenborstel kan zich niet verplaatsen tenzij je hem in een hand hebt
```

veranderingen van brush.paste:

```
brush.paste==yes → brush.paste==no : brush.wh==mouth  
// de tandpasta verdwijnt alleen van de borstel als de borstel in je mond is  
brush.paste==no → brush.paste==yes : brush.hand !=none ^ brush.wh!=mouth  
// er komt alleen tandpasta op de borstel als je hem in een hand hebt en niet in je mond
```

veranderingen van brush.hand:  
// geen restricties

1b (ii). Overgangen in het gecombineerde systeem moeten aan de volgende extra eisen voldoen:

```
tube.cap==close→tube.cap==open : brush.hand==none
tube.cap==open→tube.cap==close : brush.hand==none
// je kunt het dopje er alleen op- of afdraaien met twee handen. Inderdaad: één hand heb
// je nodig om de tube vast te houden, je kunt de tandenborstel dus niet
// tegelijkertijd vasthouden.
tube.hand== * → tube.hand==l : brush.hand !=l
tube.hand== * → tube.hand==r : brush.hand !=r
brush.hand==*→brush.hand==l : tube.hand !=l
brush.hand==*→brush.hand==r : tube.hand !=r
// je kunt niet borstel en tube tezamen in één hand houden
brush.paste==no→brush.paste==yes : tube.cap==open
// om tandpasta op de borstel te doen moet de tube open zijn.
```

2a. Een doelstelling zou ‘analyse’ kunnen zijn: hoe hangen de maximale populaties vossen resp. konijnen af van de diverse grootheden bf, br, df, dr.  
b. Het aangepaste script is:

```
bf=slider(1.5,0,5)
// birt rate of fox (horizontal)
br=slider(3,0,5)
// birth rate of rabbit (vertical)
df=slider(1,0,5)
// death rate of fox (horizontal)
dr=slider(1,0,5)
// death rate of rabbit (vertical)
dt=slider(0.15,0.01,0.2)
// how fast does time move forward?
reset=button()
// to restart the game
p=plot([gboth,gfox,grab,horLevel,verLevel])
// the graph
plf=f
// for debugging
plr=r
// for debugging
f=if((time<2)||reset,150,max(0.01,f{1}+dt*(bf*f{1}*r{1}-df*f{1})))
// lotka volterra for fox
gboth=['nPoints:50,plotType:bubble,diameter:{value:5},x:{mode:shift,ref:1},y:{mode:shift,ref:2}
',10+3*[f],10+3*[r]]
// draw the cyclic curve for the both species
gfox=['y:{mode:intp},x:{mode:shift,ref:1},width:{value:5},col_r:{value:255}',10+3*[f]]
// draw the fox population scrolling vertically
grab=['x:{mode:intp},y:{mode:shift,ref:1},width:{value:5},col_b:{value:255}',10+3*[r]]
// draw the rabbit populating horizontally
r=if((time<2)||reset,150,max(0.01,r{1}+dt*(br*r{1}-dr*f{1}*r{1})))
// lotka volterra for rabbit
time=time{1}+1
// from here to eternity
horLevel=['x:{mode:intp},y:{mode:data,ref:1}',vMake(10+3*rGauge,50)]
// a horizontal line at the height of the rabbit population size gauge
// which helps to assess the max. nr rabbits in a cycle
verLevel=['y:{mode:intp},x:{mode:data,ref:1}',vMake(10+3*fGauge,50)]
// a vertical line at the width of the fox population size gauge
// which helps to assess the max. nr foxes in a cycle
rGauge=slider(0,0,30.0)
```

fGauge=slider(0,0,30.0)

De aangepaste regels zijn rood gemaakt. Een link waarmee het model gestart kan worden is <http://www.keesvanoverveld.com/Accel/accel.htm?link=aa4780c1f5e687ae014da236775f1289> (NB: deze link is aangemaakt op 20 september 2013 en mogelijk niet meer geldig op het moment dat dit document gelezen wordt!).

2c. Het model laat nu twee extra sliders zien, waarmee een horizontale en verticale haarlijn ingesteld kunnen worden waarmee je het maximale aantal vossen resp. het maximale aantal konijnen kunt meten in een complete ecologische cyclus bij een gegeven instelling voor de respectievelijke parameters in het model. Daarmee kun je de invloed van de waarden van die parameters op de respectievelijke maxima makkelijk onderzoeken en bijvoorbeeld in een grafiek uitzetten.

## Elaboration Assignment Introduction to Modeling – Chapter 3

1a. We elaborate the tooth brush in example. The concepts are (we use abbreviations):

```
tube=[cap:{open,close},hand:{l,r,none}]
brush=[wh:{glass,mouth,else},paste:{yes,no},hand:{l,r,none}]
```

1b (i). The tube has 6 states (cap:2 X hand:3). All transitions where exactly one property changes its value are admitted, provided the following requirements are fulfilled (for brevity, we write 'transition:condition//comments' which means that the transition can only occur if the condition is true; '!=' means 'is not equal to'; '^' means 'and'; 'v' means 'or'; '\*' means: 'any value'):

Changes in tube.cap:

```
tube.cap==open → tube.cap==close : tube.hand !=none
// you cannot close the cap unless you have the tube in one hand
tube.cap==close → tube.cap==open : tube.hand !=none
// you cannot open the cap unless you have the tube in one hand
```

Changes in tube.hand:

```
tube.hand==* → tube.hand==none : tube.cap==close
// you don't put the tube down unless the cap is closed
```

The brush model has 18 states (wh:3 X paste:2 X hand:3). All state transitions where exactly one property changes its value are admitted, provided the following requirements are fulfilled:

Changes in brush.wh:

```
brush.wh==glass → brush.wh==mouth : brush.hand !=none ^ brush.paste==yes
// only put a toothbrush in your mouth when it has tooth paste on it
brush.wh==glass → brush.wh==else : brush.hand !=none
brush.wh==mouth → brush.wh==glass : brush.hand !=none ^ brush.paste==no
// don't put a toothbrush with tooth paste on it in the glass
brush.wh==mouth → brush.wh==else : brush.hand !=none
brush.wh==else → brush.wh==mouth : brush.hand !=none ^ brush.paste==yes
// only put a toothbrush in your mouth that has tooth paste on it
brush.wh==else → brush.wh==glass : brush.hand !=none ^ brush.paste==no
// don't put a toothbrush with tooth paste on it in the glass
// for all of the above transitions, we further have:
// the toothbrush cannot move unless it is in one of the hands
```

Changes in brush.paste:

```
brush.paste==yes → brush.paste==no : brush.wh==mouth
// toothpaste only vanishes from the toothbrush while it is in your mouth
brush.paste==no → brush.paste==yes : brush.hand !=none ^ brush.wh!=mouth
// toothpaste only appears on the toothbrush when it is in your hand and not in your mouth
```

Changes in brush.hand:

// no restrictions apply

1b (ii). Transitions in the combined system must fulfill the following additional requirements:

```
tube.cap==close→tube.cap==open : brush.hand==none
tube.cap==open→tube.cap==close : brush.hand==none
// you can only open or close the cap using both hands, so you cannot hold the
// toothbrush at the same time.
tube.hand== * → tube.hand==l : brush.hand !=l
tube.hand== * → tube.hand==r : brush.hand !=r
brush.hand==*→brush.hand==l : tube.hand !=l
brush.hand==*→brush.hand==r : tube.hand !=r
//you cannot hold the brush and the tube in the same hand
brush.paste==no→brush.paste==yes : tube.cap==open
// to put tooth paste onto the brush, the cap must be open.
```

2a. A purpose could be 'analyse': how do maximal populations for fox or rabbit depend on the various quantities bf, br, df, dr.

b. The adapted script reads:

```
bf=slider(1.5,0,5)
// birt rate of fox (horizontal)
br=slider(3,0,5)
// birth rate of rabbit (vertical)
df=slider(1,0,5)
// death rate of fox (horizontal)
dr=slider(1,0,5)
// death rate of rabbit (vertical)
dt=slider(0.15,0.01,0.2)
// how fast does time move forward?
reset=button()
// to restart the game
p=plot([gboth,gfox,grab,horLevel,verLevel])
// the graph
plf=f
// for debugging
plr=r
// for debugging
f=if((time<2)||reset,150,max(0.01,f{1}+dt*(bf*f{1}*r{1}-df*f{1})))
// lotka volterra for fox
gboth=['nPoints:50,plotType:bubble,diameter:{value:5},x:{mode:shift,ref:1},y:{mode:shift,ref:2}
',10+3*[f],10+3*[r]]
// draw the cyclic curve for the both species
gfox=['y:{mode:intp},x:{mode:shift,ref:1},width:{value:5},col_r:{value:255}',10+3*[f]]
// draw the fox population scrolling vertically
grab=['x:{mode:intp},y:{mode:shift,ref:1},width:{value:5},col_b:{value:255}',10+3*[r]]
// draw the rabbit population scrolling horizontally
r=if((time<2)||reset,150,max(0.01,r{1}+dt*(br*r{1}-dr*f{1}*r{1})))
// lotka volterra for rabbit
time=time{1}+1
// from here to eternity
horLevel=['x:{mode:intp},y:{mode:data,ref:1}',vMake(10+3*rGauge,50)]
// a horizontal line at the height of the rabbit population size gauge
// which helps to assess the max. nr rabbits in a cycle
verLevel=['y:{mode:intp},x:{mode:data,ref:1}',vMake(10+3*fGauge,50)]
// a vertical line at the width of the fox population size gauge
// which helps to assess the max. nr foxes in a cycle
rGauge=slider(0,0,30.0)
fGauge=slider(0,0,30.0)
```

Adapted lines have been indicated in red. A link to run the model is <http://www.keesvanoverveld.com/Accel/accel.htm?link=aa4780c1f5e687ae014da236775f1289> (Notice: this link has been made on September 20, 2013; it may have expired at the time of reading this document!).

2c. The model now shows two additional sliders, allowing to position a horizontal and vertical hairline. This helps to measure the maximal number of fox and rabbit for an entire ecological cycle, given a setting for the respective parameters in the model. This facilitates investigating the influence of the values of these parameters, possibly plotting them in a graph..

