Lab Assignment 11

An Ecosystem with Multiple Trophic Levels

Written by

Prof. Erin N. Bodine

Prepared for

Math 214: Discrete Math Modeling (Fall 2014)

Rhodes College

PreLab Assignment: Done in class on Tuesday, 11/20/2014

Lab Assignment Due: Tuesday, 11/25/2014 prior to class

In Lab 8, we introduced and explored the Rabbits ABM which was simulated in `Rabbits.nlogo`.

Description of the Rabbits ABM: This agent-based model explores a simple ecosystem made up of rabbits and grass. The rabbits wander around randomly, and the grass grows randomly. When a rabbit bumps into a patch of grass, it eats the grass and gains energy. If the rabbit gains enough energy, it reproduces. Each rabbit loses energy every time it moves and reproduces. If a rabbits loses too much energy it dies.

Now, that we are gaining proficiency with writing code in NetLogo, we will modify the Rabbits ABM to simulated the more complex ecosystem illustrated in Figure 1. This systems is modeled after the dynamics described in [1]. Skinks are a type of lizard native to New Zealand's dry grasslands, however, after the introduction of European rabbits (*Oryctolagus cuniculus*) many of the native skink species started to decline. The rabbits and the skinks share a common predator, ferrets. The rabbits are the primary food source for the ferrets, and thus when the rabbit population increases, there is a corresponding increase in the ferret population. However, when the rabbit population decreases, the ferret population preys on skinks with increasing frequency, which can have devastating effects on the skink population if the ferret population is still large during a rapid rabbit population decline. Thus, there is a form of competition, known as ***apparent competition***, between the rabbits and the skinks occurring (mediated through the ferret predators).

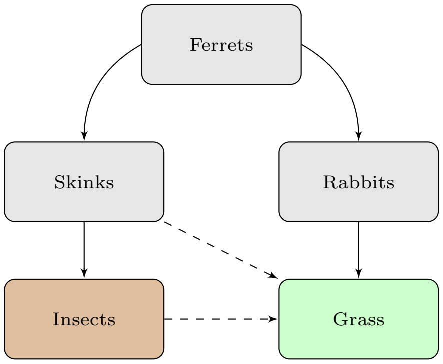

Figure 1: Illustration of ferret-skink-rabbit food web. Solid lines indicate food source; dashed lines indicate habitat use.

(a) Endangered Otago skink.

(b) European rabbit.

Figure 2: Prey sources for New Zealand ferrets.

There is however, another layer of complexity. Both the skinks and their food source (a variety of insects), use the rabbits main food source, grass, as their habitat. When the rabbit population increases to an extremely high density, it ravages the grassland habitat of the skinks and their food source. This has a two-fold effect. It reduce the abundance of the skinks food source, and it removes natural cover that would afford some measure of protection from predators like ferrets.

In this lab, we will use an ABM to simulate the complex interactions within this ferret-rabbit-skink food web, and use the results of our simulations to draw some conclusions about the system. For the prelab, we will focus on adding the ferret predators to the existing rabbits and grass ecosystem represented in `Rabbits.nlogo`. For the lab assignment, we will add the interactions with the skinks and their food source.

NetLogo Skills

At this point, you already have most of the skills that you need to construct this ABM. However, we have yet to implement a model with multiple types of agents. In NetLogo, all agents are referred to as turtles, but if there are multiple types of agents we can define different *breeds* of turtles. We define the different breeds at the beginning of the code using the command `breed` (see NetLogo dictionary entry). For example, in the Rabbits.nlogo code, the breed rabbits is defined. Note you must define the words used to refer to the entire group of agents of that breed (in this case `rabbits`), and the word used to refer to a single agent of that breed (in this case `rabbit`). See the first non-commented line from snippet of the `Rabbits.nlogo` code below.

`Snippet of Rabbits.nlogo`

```
;---------------------------------------------------
;----- Create Agents -------------------------------
;---------------------------------------------------
breed [ rabbits rabbit ]
rabbits-own [ energy ]
```

Once a breed is created, it can have unique variables defined for that breed only. Thus, in the example above, the agent variable `energy` is defined only for the breed `rabbits`. If there exist agents that are not of the breed `rabbits`, then they will not have an `energy` variable.

We will want to modify the `Rabbits.nlogo` code to include the ferret population as a second breed agents. Both sets of agents will have the agent variable `energy`, but the ferret population will additionally have an agent variable called `num-rabbits-eaten`. To create this structure in our new ABM, we would use the following snippet of code.

`Snippet of Ecosystem.nlogo`

```
;---------------------------------------------------
;----- Create Agents -------------------------------
;---------------------------------------------------
turtles-own [ energy ]

breed [ ferrets ferret ]
ferrets-own [ num-rabbits-eaten ]

breed [ rabbits rabbit ]
rabbits-own [ ]
```

Notice the `turtles-own` primitive is used to create the agent variables that all agents will have, while the `<breeds>-own` primitive is used to create the agent variables unique to a particular breed. Note that in the code above, the rabbits do not have any agent variables that are unique to their breed. In fact, we could omit the last line of code: `rabbits-own [ ]`.

If we want to access a particular breed, we could ask that breed to execute a series of commands using the `ask` primitive.

```
ask <breeds> [ ...  ]
```

Thus, for example, we could ask the ferrets to set the `num-rabbits-eaten` variable to 0 by using the command

```
ask ferrets [set num-rabbits-eaten 0]
```

In some contexts you might want to determine the breed of a particular turtle. Once breeds are established, each turtle has a built-in variable called `breed` which defines its breed. For example, if you want to check the breed of the agent on a given patch you could use the command

```
ask patches with [any? turtles-here] [print breed]
```

In the ferrets-rabbits-skinks ecosystem example, if you wanted the grass on any patch visited by a rabbit to be "eaten" (i.e. turned to black), you could use the command

```
ask patches [if any? turtles-here with [breed = rabbits] [set pcolor black] ]
```

You now should be able to finish the following pre-lab and lab assignments.

PreLab Assignment

1. From Moodle, download the `Rabbits.nlogo`, and save it as `Lab11_YourLastName.nlogo`.
2. Add a breed called ferrets. Give this breed their own agent variable called `num-rabbits-eaten`. Both the rabbit and ferret breeds should have an agent variable called `energy`.
3. Modify the SETUP procedure to create an initial number of ferrets as determined by a global variable (`initial-num-ferrets`) defined on the interface tab. You can use a slider that ranges between 0 and 100. Each ferret should be the same shape (your choosing) and color (your choosing), be placed in a random location within the model world, and be given a random integer-valued initial starting energy in the range $[0, 9]$.
4. Modify the GO procedure to do the following:
   (a) If there are not any rabbits or ferrets, the GO procedure should stop.
   (b) All the patches execute the GROW-GRASS procedure.
   (c) All the agents (turtles) execute the MOVE procedure.
   (d) All the rabbits execute the EAT-GRASS, REPRODUCE, AND DEATH procedures.
   (e) All the ferrets execute the EAT-RABBITS, REPRODUCE, AND DEATH procedures. You will create the EAT-RABBITS procedure shortly.
5. Create an EAT-RABBITS procedure. If there are any rabbits in a 2 patch radius of the ferret currently executing the EAT-RABBITS procedure, then the following should occur:
   (a) Use the command

   ```
   let rabbit-prey one-of rabbits in-radius 2
   ```

   to select one of the rabbits on the current patch, and save that agent to the local variable `rabbit-prey`. To fully understand this command, you should look up the primitives `one-of` and `in-radius`.

   (b) Set the `energy` variable of the ferret calling the EAT-RABBITS procedure to its current energy plus the energy of the `rabbit-prey`.

   (c) Kill the agent `rabbit-prey`. Look up the primitive `die` in the NetLogo Dictionary.
   (d) Increase the ferret variable `num-rabbits-eaten` by 1.

6. All other procedures should remain the same.

7. Modify the Populations plot on the interface to include a count of the number of ferrets at each time step.

8. Make sure you save your program.

9. Set the global variables so that `initial-num-rabbits` is 325, `initial-num-ferrets` is 50, `birth-threshold` is 4, `grass-energy` is 5.5, and `grass-grow-rate` is 0.033. Run the model several times.
   (a) Take a screenshot of a simulation where the ferrets and rabbits die off fairly quickly.
   (b) Take a screenshot of a simulation where the ferrets die off quickly but the rabbits and grass eventually approach an equilibrium.
   (c) Take a screenshot of a simulation where the ferrets, rabbits, and grass oscillate is size for some time before the ferrets and rabbits eventually die out.

10. Submit your NetLogo code and the three screenshots to Moodle.

Lab Assignment

1. Start with the file you submitted for the PreLab assignment. If there were any comments I gave you about fixing your code, implement them now.

2. Add another breed called `skinks`. The skinks will have an `energy` variable like the other two breeds, but no other additional variables.

3. Add another variable to the ferrets breed called `num-skinks-eaten`.

4. Modify the SETUP procedure to create an initial number of skinks as determined by a global variable (`initial-num-skinks`) defined on the interface tab. You can use a slider that ranges between 0 and 300. Each skink should be the same shape (your choosing) and color (your choosing), be placed in a random location within the model world, and be given a random integer-valued initial starting energy in the range $[0, 9]$.

5. Modify the GO procedure so that:
   (a) If a skink is on a patch of grass it does not move and it gains half of the `grass-energy` of the patch. Note that a skink gaining energy from a patch of grass does not deplete the grass from the patch nor lower the energy of the patch since technically the skinks are eating insects occupying the habitat the grass provides. To keep this model simple, we are not explicitly modeling the insects.
   (b) If a skink is not on a patch of grass it should execute the move procedure.
   (c) After all agents have moved, the skinks should execute the REPRODUCE and the DEATH procedures, then the rabbits should execute the EAT-GRASS, REPRODUCE, and DEATH procedures, then the ferrets should execute the EAT-PREY, REPRODUCE, and DEATH procedures.

6. Modify the EAT-RABBITS procedure to become the EAT-PREY procedure. If there are any agents with the breed `rabbits` or `skinks` in a 2 patch radius, the ferret should
    (a) Use the command

    ```
    let prey one-of turtles with [breed = rabbits or breed = skinks] in-radius 2
    ```

    to select on of the rabbits or skins in a 2 patch radius to the ferrets prey.

    (b) Set the `energy` variable of the ferret calling the EAT-PREY procedure to its current energy plus the energy of the `prey`.

    (c) If the ferret just ate a rabbit, increase the `num-rabbits-eaten` variable by 1. If the ferret just ate a skink, increase the `num-skinks-eaten` variable by 1.

    (d) Kill the agent `prey`.

7. All other procedures should remain the same.

8. Modify the Population plot on the interface to include a count of the number of skinks at each time step.

9. Make sure you save your program.

10. Set the global variables so that `initial-num-rabbits` is 325, `initial-num-ferrets` is 50, `initial-num-skinks` is 150, `birth-threshold` is 4, `grass-energy` is 5.5, and `grass-grow-rate` is 0.033. Run the model several times.

    (a) Take a screenshot of a simulation where the ferrets, rabbits, and skinks die off fairly quickly.

    (b) Take a screenshot of a simulation where the skinks die off quickly but the ferrets, rabbits and grass oscillate in size for some time before the ferrets and rabbits eventually die out.

    (c) Take a screenshot of a simulation where there is a skink population explosion.

11. Submit your NetLogo code and the three screenshots to Moodle.

Note, this is the last Lab Assignment. However, for the final project we will be expanding on this ABM and using discrete difference equations, and matrix models to further investigate this ferret-rabbit-skink system.

References

[1] G. Norbury. Conserving dryland lizards by reducing predator-mediated apparent competition and direct competition with introduced rabbits. 2001. *Journal of Applied Ecology*, 38(6): 1350-1361.