EXERCISE 4b: BUILD EXPRESSIONS WITH FUNCTIONS

Functions provide most of the raster processing capabilities for Map Algebra. There are 168 of them, as opposed to only 29 operators. Each function is a predefined model that returns a computed value; you treat functions as objects in an expression.

It is not practical to present all of the functions in this class. However, they all follow the same syntax rules and use one of four methods to compute results, so learning to use a new function just requires reading the documentation and applying the standard syntax rules. In this exercise, you will work with functions from each processing group: local, focal, block, zonal, and global.

STEP 1: USE LOCAL FUNCTIONS IN EXPRESSIONS

In this step, you will use several local functions. These process each cell individually. First, start ArcMap and load the map you saved in Exercise 5A.

- □ Start *ArcMap* and open … *Exercise05\MapAlgebra.mxd*.

- □ Make these Environment Settings:
  - *Current Workspace: ... \Exercise05*
  - *Scratch Workspace: ... \Exercise05*
  - *Output Extent: Same as Layer "Elevation"*
  - *Cell Size: Same as Layer "Elevation"*

Your first task will be to select a random sampling of cells from your Elevation raster. This technique is useful for several types of statistical analysis. You will use RAND to generate random cell values and then use them to select Elevation cells.

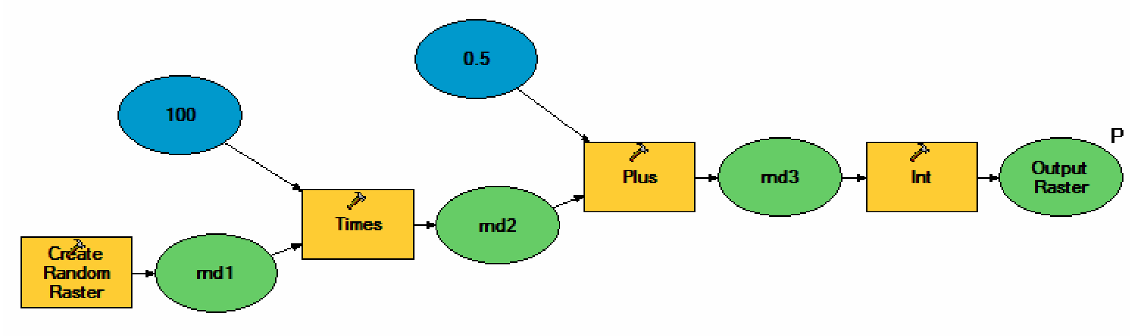
The RAND function generates a random fractional value in the range of zero to one for each cell. For this example, you will rescale the range by multiplying the random numbers by 100. Then you add 0.5 to the decimal values to round them up and finally truncate them to integers with the TNT function. This process could be a series of steps for each operation or one nested Map Algebra expression like the one shown below:

INT (RAND( ) * 100 + 0.5)

This process has been implemented as a model so that you can see how you would use the Map Algebra functions and operators as tools in a model. Now you will add a toolbox to ArcToolbox, review the model, and run it.

- □ In the *ArcToolbox*, right-click in the background and select *Add Toolbox*.

- □ In the Add Toolbox dialog, navigate to and open the toolbox ... *Exercise05\X5*.



- ☐ In the Arc Toolbox, expand the X5 toolbox to see the *MakeRandom* model.
- ☐ Right-click on *MakeRandom* and select *Edit* to open the model.

The orange boxes are ArcToolbox tools. The blue ovals are parameters used by the tools. The green ovals are the output rasters; all but the last are intermediate and will be deleted automatically after the model runs. You will learn how to build models with the Model Builder in a later lesson.

- ☐ In the *MakeRandom* window, click *Model* > *Close*. Do not save any changes.

You run models like you do any other ArcToolbox tool.

- ☐ *Run ..> X5> MakeRandom:*
  - Output Raster: type **RndGrd**

- ☐ Right-click *rndgrd* and click *Open Attribute Table*.

The table shows that all values from 0 to 100 are represented and that there are about the same number of cells for each value. The important thing about this raster is that cells with the same value are randomly dispersed spatially. By selecting values from this random raster, you can create a random sampling of the values in another raster for any percentage you like, from 1 to 100 percent.

- ☐ Close the *Attributes of Random* table.
- ☐ Turn off *Random* and close its legend.

Of the dozens of local functions, CON is perhaps the most important because it is used to create conditional statements. Now use the CON function to create a one percent random sampling of the Elevation layer.

- ☐ *Run .. > Map Algebra > Single Output Map Algebra:*

□ MapAlgebra expression: type

CON (rndgrd eq 1, Elevation)

□ Output raster: type **Samples**

□ Turn off all layers except samples so you can see it.

The speckles you see in the map are cells representing a random one percent sampling of the Elevation raster. You could use these random values in further statistical analysis. In the CON expression, if the rndgrd raster equals 1, a cell value from the Elevation raster is output; otherwise, the cell is output as NoData.

□ Turn off samples and collapse its legend.

Turning cells into NoData and testing for the presence of NoData are two of the more important tasks in Map Algebra. You have just seen one technique for setting cells to NoData: omitting the {false_expression} argument in the CON function.

Now you will use the SETNULL function to create a mask, where non-Forest Service land and water bodies are set to NoData, then you will use the mask to clip the Soil layer.

□ Turn on *Soil and Ownership*.

First you will create the mask with the SETNULL in the Single Output Map Algebra tool. For variety, you will use the algebraic symbols for the Not Equal (NE or <>) operator and the Equal (EQ or = =). Be careful to type two equal sign characters for the Equal operator.

□ Run Single Output Map Algebra:

□ Map Algebra expression: type

SETNULL(Ownership < > 200 OR Soil = = 901, 1)

□ Output raster: type **SoilMask**

□ Turn off all layers except *soilmask* so you can see it.

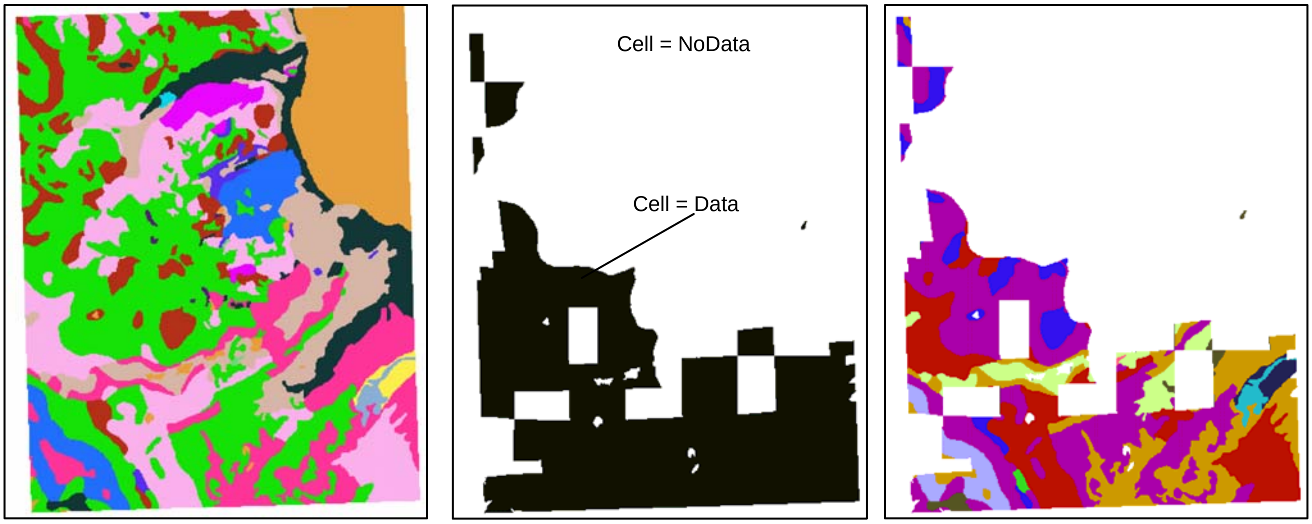
The dark areas have a value of one. All other cells (non-Forest Service land or water bodies) have been turned into NoData. Now use the mask to clip your soils.

□ *Run ... > Extraction > Extract by Mask:*

□ Input raster: select *Soil*

☐ Input raster or feature mask data: select *SoilMask*

☐ Output raster: type *ClipSoil*



Soil SoilMask ClipSoil = Soil

NOTE: If you had set SoilMask in the mask environment, you could have clipped the Soil raster by just typing its layer name in the Single Output Map Algebra tool.

The clipsoil layer only has soil data for Forest Service properties that are not water. This technique is useful for clipping rasters to an irregular outline, like extracting data for a specific city from a countywide database.

☐ Turn off all layers and close their legends.

STEP 2: USE FOCAL FUNCTIONS IN EXPRESSIONS

The focal functions create neighborhoods around each cell as it is being processed and use the values of the neighboring cells to compute a value for the processing cell (e.g., for a cell, add up the values of its eight neighbors and write the sum to the output cell). All of the focal functions allow you to define the size and shape of the neighborhood.

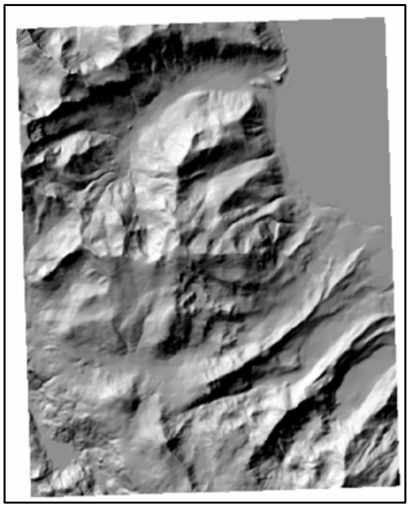
Most of the focal functions compute a statistic for the neighborhood, like the sum, min, max, mean, or standard deviation. These are best applied to continuous data, like distance or elevation. Others, like variety and majority, may be used with discrete data (e.g., How many plant species are within 1000 meters of each cell?).

In the following sequence, you will use FOCALMEAN and FOCALMEDIAN to smooth your Elevation surface before you generate a Hillshade from it. This is a filtering technique that may be used to soften the appearance of a Hillshade.

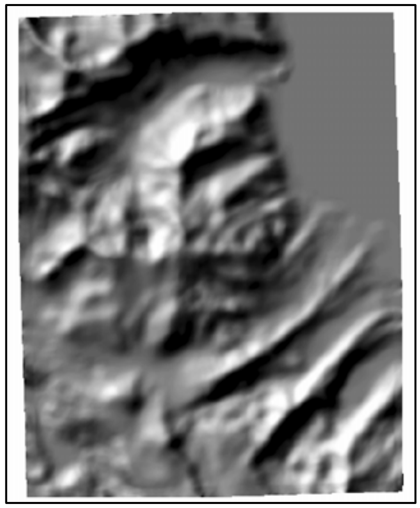
To save processing time, you will nest the Hillshade and focal functions.

□ Add ... *Exercise05\Hillshade.lyr* to your map.

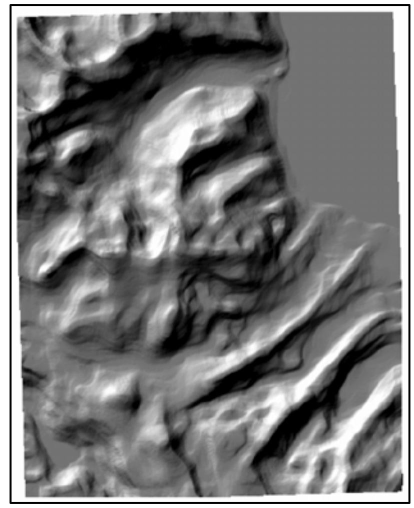
□ Turn on Hillshade. Notice that the topography is sharply defined.



Hillshade
(no filter)



Hillshade
(Focalmean)



Hillshade
(Focalmedian)

FOCALMEAN writes the mean—or average—value within the neighborhood to the output cell. As applied to elevation, it can be used to smooth or soften the surface.

□ Run Single Output Map Algebra:
□ Map Algebra expression: type
  - **HILLSHADE (FOCALMEAN (Elevation, Circle, 6))**

□ Output raster: type **HillMean**

NOTE: You could use the Focal Statistics tool with the MEAN statistic (from the Neighborhood toolset) followed by the Hillshade tool (from the Surface toolset) to do the same thing. However, using the Single Output Map Algebra tool combines both operations in one expression.

□ Turn *HillMean* on and off, comparing it to the *Hillshade* layer.

The HillMean layer shows that FOCALMEAN removed much of the detail from the elevation data. You may use this technique to build a generalized relief map for cartography (it is commonly called the "Swiss method") if you want to keep the details of the surface from interfering with other data, like land use, that is drawn on top of it.

FOCALMEDIAN writes the median value within the neighborhood to the output cell. The median is that value that has the same number of values above and below it. As applied to elevation, it can be used to emphasize areas of change in the surface.

□ Turn off *HillMean* and collapse its legend.

- ☐ *Run Single Output Map Algebra:*

- ☐ Map Algebra expression: type

  - **HILLSHADE (FOCALMEDIAN (Elevation, Circle, 6))**

- ☐ Output raster: type **HillMed**

- ☐ Turn *HillMed* on and off, comparing it to the *Hillshade* layer.

The HillMed layer shows that FOCALMEDIAN has emphasized areas of change in the elevation data. These are the terraces, or stairsteps, that are visible in the graphic.

- ☐ Turn off all layers and collapse their legends.

Three of the focal functions (FOCALMINORITY, FOCALMAJORITY, and FOCALVARIETY) are appropriate for use with coded data. In the following sequence, you will use FOCALVARIETY to find the number of different vegetation types within a 6-cell radius (about 600 feet) of every cell. This is a measure of vegetation diversity and could be used to define critical habitats.

- ☐ Turn on *Landcover* and open its legend. Briefly review the code descriptions.
- ☐ *Run ... > Neighborhood> Focal Statistics:*
- ☐ Input raster: select *LandCover*
- ☐ Output raster: type **vegvar**
- ☐ Neighborhood: select *Circle*
- ☐ Radius: type **6**
- ☐ Statistics type: select *VARIETY*

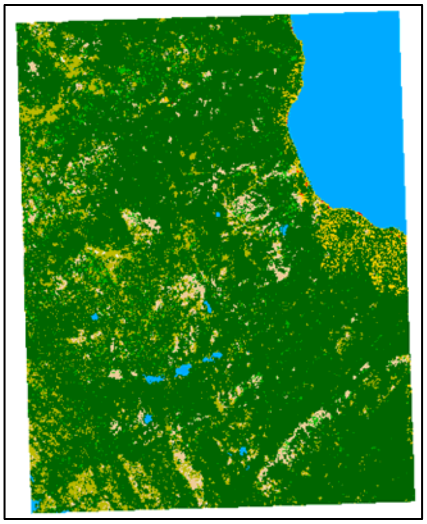
NOTE: This will be equivalent to the Map Algebra expression:
FOCALVARIETY(Landcover, Circle, 6)

The cells with the highest values have the greatest number of vegetation types within the relatively small neighborhood around them. The new layer shows the variety of LandCover instead of vegetation because you did not turn nonvegetation cells in the LandCover layer into NoData before running FOCALVARIETY.

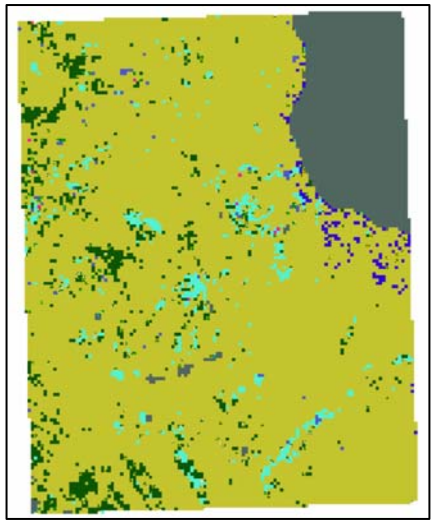
- ☐ Turn off all layers except LandCover and collapse their legends.

STEP 3: USE BLOCK FUNCTIONS IN EXPRESSIONS

The block functions are often used to generalize data. They are similar to the focal functions, except that the neighborhoods are applied within blocks of cells, which are in turn offset during processing so that no block overlaps another. The value computed within the neighborhood of a block is written to every output cell that falls within the block. Now use BLOCKMAJORITY to generalize the LandCover layer to 90-meter square areas (a 3x3 cell block).



LandCover



LandGen

- ☐ Run ... > Neighborhood> Block Statistics:
- ☐ Input raster: select *LandCover*
- ☐ Output raster: type **LandGen**
- ☐ Neighborhood: select *Rectangle*
- ☐ Height: type **3**
- ☐ Width: type **3**
- ☐ Statistics type: select *MAJORITY*

NOTE: This will be equivalent to the Map Algebra expression: BLOCKMAJORITY ([LandCover], RECTANGLE, 3, 3)

- ☐ Right-click *LandCover* and select *Copy Raster Symbology*.

- ☐ Right-click *LandGen* and select Paste *LandCover Symbology*.

- ☐ Turn *LandGen* on and off, comparing the graphic to the *LandCover* layer.

As you can see, BLOCKMAJORITY has generalized the land cover data to larger areas. The important thing about the block functions is that they perform a type of thematic generalization, not spatial. If you checked th properties of the LandGen layer, you would see that it has the same 30-meter cell size as Landcover.

- ☐ Turn off *LandGen* and close its legend.

STEP 4: USE ZONAL FUNCTIONS IN EXPRESSIONS

Most of the zonal functions compute statistics for measured data and summarize the statistic by the zones defined in a coded raster. For example, you might define the mean elevation for each land cover class. A zone is defined as all cells that have the same value, regardless of whether they are connected or not. For example, every cell that has a code of Residential belongs to the same Residential zone. Now use the ZONALMEAN function to compute the mean elevation within each land cover class.

- □ *Run ... > Zonal> Zonal Statistics*:

- □ *Input raster* or feature zone data: select *LandCover* (This is the dataset that defines the zones.)

- □ Zone field: select *Value* (This is the field that holds the values that define each zone.)

- □ Input value raster: select *Elevation* (This is the raster whose values you want to summarize.)

- □ Output raster: type **ElevLand**

- □ Statistics type: select *MEAN*

NOTE: This will be equivalent to the Map Algebra expression:
ZONALMEAN (LandCover, Elevation)

The results are not immediately obvious from the map. ArcMap assumes that floating-point rasters should be symbolized with a stretched renderer, making the zone pattern hard to see. If you change to a unique value renderer, the patterns become apparent.

- □ Change the symbology of the *ElevLand* layer to *Unique Values*.

You should now see the same number of classes that are in the LandCover layer. The legend for ElevLand and shows the mean elevation for each zone.

- □ Turn off all layers and collapse their legends.

While the zonal functions are useful in Map Algebra (e.g., you might embed ZONALMAJORITY in a CON function), they are not well suited for simply looking at the statistics. For that, you can use the ZONALSTATS function to return a table of the desired statistic (MIN, MAX, ALL, or so on). You will now use it to compute all the zonal statistics for LandCover and Elevation and write them to a dBASE file.

- □ *Run ... > Zonal > Zonal Statistics as Table:*
- □ *Input raster* or feature zone data: select *Landcover*
- □ Zone field: select *Value*
- □ Input value raster: select *Elevation*

□ Output table: type **Landstats.dbf**

This will be equivalent to the Map Algebra expression:
ZONALSTATS (LandCover, Elevation, ALL)

□ In the ArcMap Table of Contents, click the *Source* tab.

□ Open the *LandStats* table.

The Value field contains the LandCover codes. The Area field contains the computed area of each zone, expressed in square map units (meters, in this case). The other fields have statistics about the elevation values found in each zone. The Variety, Majority, and Minority statistics are only useful if the input value layer has coded data like soils. You can view all the statistics easily in the table, but you cannot use them for analysis because they are not associated with a raster layer. You can fix this by using ArcMap to join the new table to your LandCover attribute table.

□ Close the *Attributes of LandStats.dbf* window.

□ In the ArcMap *Table of Contents*, click the *Display* tab.

□ Right-click *LandCover* and select Joins and Relates > Join.

□ Choose the field in this layer: select *Value*

□ Choose the table to join: select *LandStats*

□ Choose the field in the table: select *VALUE.*
NOTE: Click Yes to create an index to the join field.

□ Right-click *LandCover* and select *Open Attribute Table*.

The Attributes of LandCover table now includes the fields from the LandStats.dbf file. You could use the joined fields for performing cell selections based on attributes or as the basis for symbol assignments, but you cannot use them in Map Algebra.

□ Close the Attributes of *LandCove*r table.

□ Turn off all layers and collapse their legends.

Several of the zonal functions return a geometric measure (area, perimeter, centroid, and thickness) of the input zones. Their names all reflect the measure they return (e.g., ZONALAREA) and all of them only require an input zone layer.

□ Add ... *\Exercise05\AirportSites.lyr* to your map.

□ Turn on *AirportSites* and open its legend. Review the site IDs.

The AirportSites layer shows four candidate sites for a general aviation airport. Each site is a zone, and they are uniquely numbered from Ito 4. The site you select must be at least 1,600 meters long (one mile) to accommodate the runway and must have an east-west orientation because of local wind conditions. Now use ZONALCENTROID to measure the major and minor axes of the ellipse that best represents each zone.

- □ *Run ... > Zonal > Zonal Geometry*:
- □ *Input raste*r or feature zone data: select *AirportSites*
- □ Zone field: select *Value*
- □ Output raster: type *Airzones*
- □ Geometry type: select *CENTROID*
- □ Output cell size: type **30** (or accept the default)
  NOTE: This will be equivalent to the Map Algebra expression:
  ZONALCENTROID (AirportSites)

- □ In the map, zoom in on the smallest *AirportSite* so it fills the display.

If you look very closely, you will see that the function returned a single cell for each zone. They represent the centroid of each zone. The most valuable results of ZONALCENTROID are in the output raster's attribute table.

- □ Right-click *AirZones* and select *Open Attribute Table*

From the table, you see that only zone I satisfies the criteria. Its major axis is 2,041) meters long and it has a genera) east-west orientation of 20.8 degrees (the orientation angles increase counterclockwise from zero degrees at due east).

- □ Close the Attributes of *AirZone* window.

- □ Turn off all layers and collapse their legends.

STEP 5: USE GLOBAL FUNCTIONS IN EXPRESSIONS

The global functions compute an output raster, where the value at each cell is potentially a function of all the cells in the input raster(s). Most of them compute distances (like measuring the least-cost path from each cell to a shopping center) or analyze surface hydrology (like determining the flow of water over a surface). These are presented in detail elsewhere, so this step will present the only two global functions that do not fit in the above categories: SLICE and REGIONGROUP.

SLICE is a classification tool that divides the input data into ranges. You will use it to classify your elevation layer into zones based on the actual range of terrestrial elevations: -396 meters (the elevation of the Dead Sea) to 8,848 meters (the top of Mt. Everest). You will request that this range be divided into 100 zones, but SLICE will only return zones within the -396 to 8,838 meter range. You will enter the function as an expression in the Single Output Map Algebra tool.

- □ *Run ... > Map Algebra > Single Output Map Algebra:*
- □ Map Algebra expression: type
  SLICE(Elevation, EQINTERVAL, 100, 1, -396, 8848)
- □ Output raster: type **ElevZone**

The legend for ElevZone shows values of 25 to 34. From a possible range of Ito 100, this indicates that the study area is low relative to the earth as a whole. Using this technique, you could classify many elevation layers to ensure that symbology across the layers would be consistent.

- □ Turn off all layers and collapse their legends.

REGIONGROUP groups connected cells into zones and assigns an ID to each new zone. Cells may be grouped based on their attributes (e.g., connected residential cells) or spatially (e.g., cells bounded by freeways).

In the last exercise, you modeled locations for a tree farm. Toward the end, you used the CAND operator to associate soil types to the GoodFarm sites you had identified. This did not produce the desired result because even though there were several distinct sites, they all had the same ID and you could not tell which soils went with which sites. You will revisit that problem, but this time you will use REGIONGROUP to assign IDs to each farm before you use CAND to associate soils with each one.

- □ Turn on *goodfarm* and open its legend. Note that the background has a value of 0.

Recall that values of one are potential farm areas and values of zero are non-suitable areas. Now use a compound Map Algebra expression to simultaneously assign IDs to each site and turn the background values of zero into NoData.

- □ Run Single Output Map Algebra:

- □ Map Algebra expression: type
  CON(GoodFarm < > 0, REGIONGROUP(GoodFarm))

- □ Accept the default name and location of the Output raster

- □ Rename the new raster layer *FarmSite*

Notice that there are twelve zones in the legend. They are numbered 2-13 because the background (0) became zone 1, which was turned into NoData by the CON function. This is the simplest and most commonly used syntax for REGIONGROUP; that is, you provide a raster layer as its only argument. It then finds groups of connected cells that have the same attribute, turns them into regions, and assigns unique IDs to them. Now complete the task by using CAND to associate soils with each farm site.

- ☐ *Run ... > Math > Logical> Combinatorial And:*
- ☐ *Input raster* or constant value 1: select *FarmSite*
- ☐ *Input raster* or constant value 2: select *Soil*
- ☐ Output raster: type **FarmDirt**

NOTE: This will be equivalent to the Map Algebra expression:
FarmSite CAND Soil

You called the output FarmDirt because you created a FarmSoil raster in the last exercise and you cannot overwrite it. Now inspect the attributes of the new layer.

- ☐ Open the attributes of FarmDirt.

From the table, you can see which soils are found within each farm site. To make it easier to read the table, you could join the soil descriptions from the Attributes of Soil table using the technique you learned earlier. Then you could choose between the sites based on the mix of soils in each.

- ☐ Close the Attributes of *FarmDirt* table.

- ☐ Turn off *GoodFarm* and *FarmDirt* and close their legends.

This concludes Exercise 5B. Many operators, functions, and techniques have been presented; but the goal of the exercise was to help you start thinking about problem solving in the context of Map Algebra.

You have also seen how the Spatial Analyst Map Algebra operators and functions can be used in Map Algebra expressions, as tools in ArcToolbox, and even in models.

- ☐ Save your map.
- ☐ Exit ArcMap.

EXERCISE END