# Map Algebra

# Lesson 4A overview

- Introduction to Map Algebra (4A)
  - Language components
  - Syntax and rules
  - Objects
  - Operators
  - Commands
  - Exercise 5A
- Map Algebra functions (4B)
  - Function syntax
  - Local functions
  - Focal and Block functions
  - Zonal functions
  - Global functions
  - Exercise 5B

# Map Algebra: The language of raster

- **A data-manipulation language designed for raster**
  - **Math-like expressions**

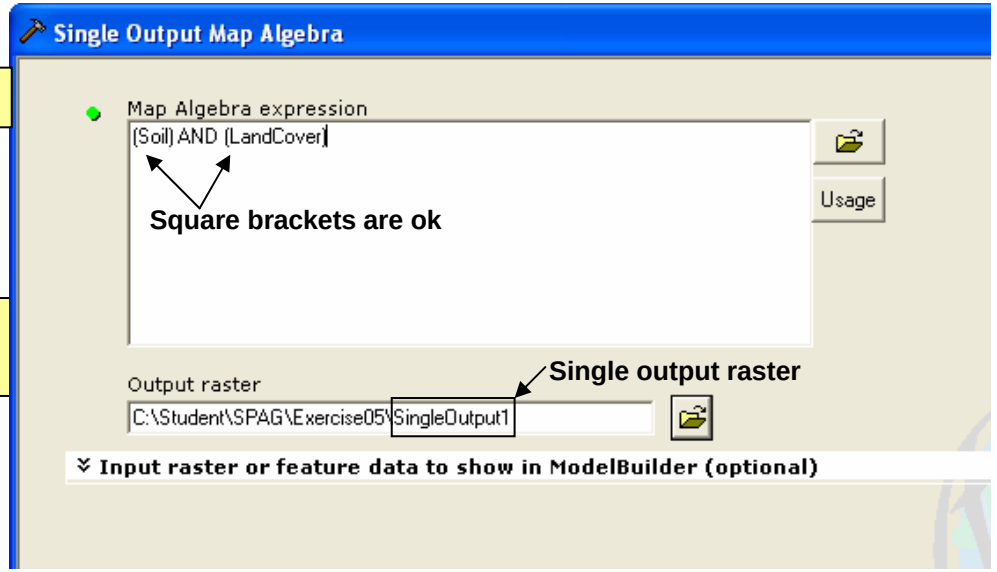
```
AqSuit = (SoilSuit * 0.75) ÷ (SlpSuit * 0.25)
```

- **Parts of the language:**
  - **Objects - Raster, vector, numbers, constants, variables...**
  - **Operators – “+”, “-” ,“/” ,“GT” ,“LE”, “AND”, “OR”**
  - **Functions - SLOPE, FOCALMEAN, SIN,**
    - **Central part of the language; over a hundred**
    - **User interface implements some, like SLOPE**
  - **Rules - For building expressions and using functions**

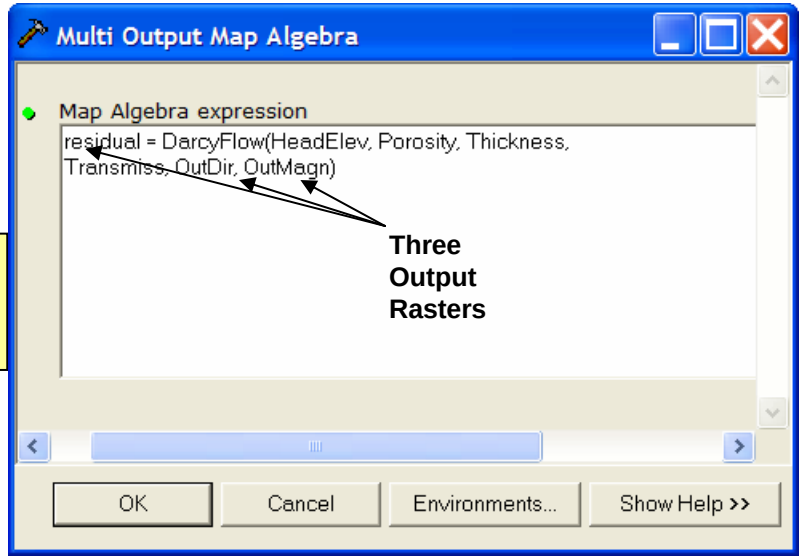
# Map Algebra geoprocessing tools

Single Output Map Algebra

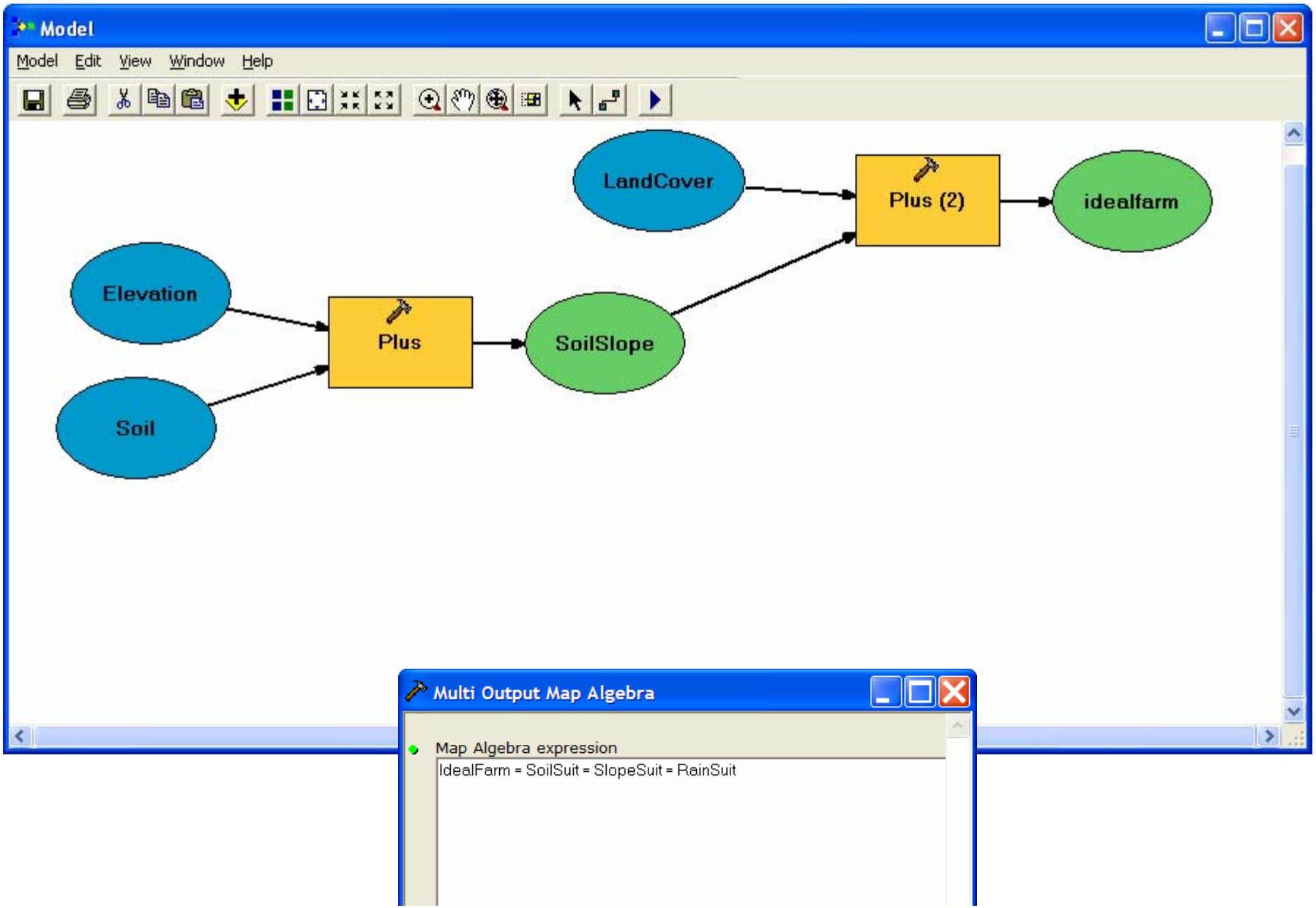
Output path:
Appears on the tool




Multi Output Map Algebra
Output path:
Set in Environment Settings dialog



# Geoprocessing tools vs Model Builder



# Expression syntax rules

- Delimit operators and objects with blanks

```
Wrong: Layer+Layer2+Layer3
Right: Layer1 + Layer2 + Layer3
```

- Operators are evaluated by precedence level

```
Layer1 + Layer2 * Layer3
```

- Override operator precedence with parentheses

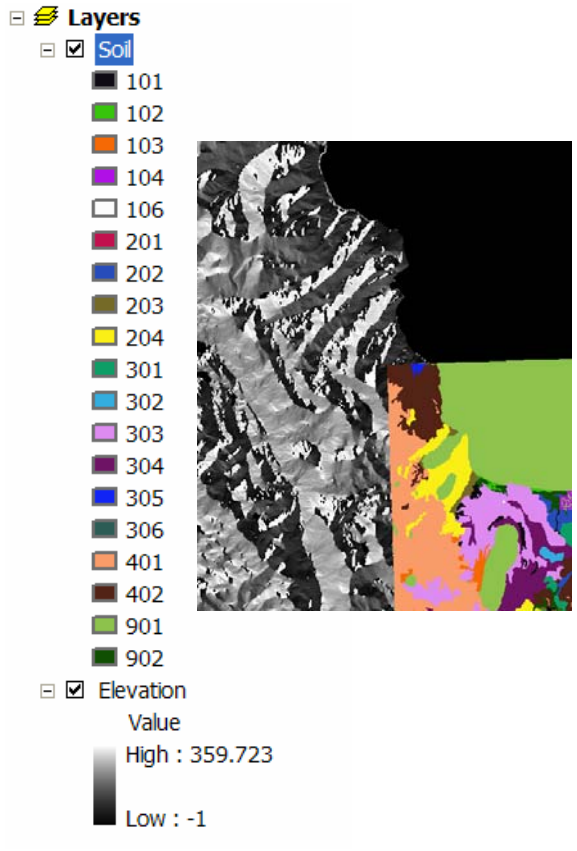
```
(Layer1 + Layer2) * Layer3
```

- Nested parenthetical expressions evaluate first

```
(Layer1 + Layer2) / 4 - Layer3
```

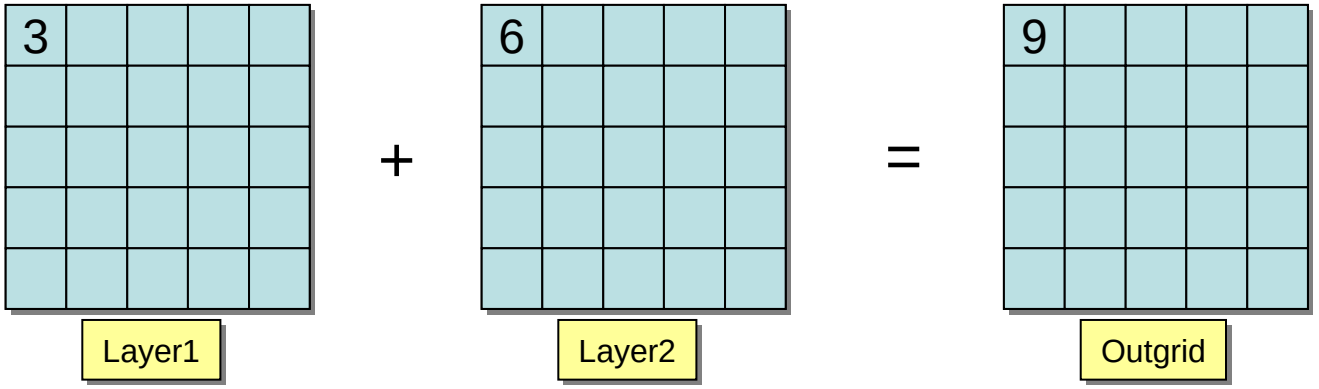
# Expression results

- Expressions return grids, vector data, tables, etc.
  - Depends on functions used
  - Most expressions return grids
- Tools supply default output names
  - Can rename
- Layers added to ArcMap Table of Contents



# Expression evaluation

- How expressions are processed




- 1. Define an empty output *raster* based on the analysis environment
- 2. Position to the next *output* cell (start at row 0, column 0)
- 3. Resample input raster(s) to determine corresponding cell values
- 4. Evaluate the expression and write the result to the output cell
- 5. Repeat steps 2—4 for all output cells

# Map Algebra objects

- **Objects used in expressions or as function arguments**

  - **Raster as layers or paths**

    ```
    SLOPE (Elevation)
    SLOPE (a:\student\database\arelev)
    ```

  - **Vector as paths**

    ```
    SHAPEGRID (a:\student\database\avroad.shp)
    ```

  - **Tables - Output by a few functions**
  - **Numbers - Integer or decimal**
    - **Scientific notation okay ($1.234e^2$ is the same as 123.4)**
  - **Constants and variables (P1, DEG, $$ROWMAP, $$NROWS, etc.)**
    - **Built in for your convenience**

# User attributes in expressions

- You may use numeric VAT fields in expressions
- Reference with *Layer.field* notation

Vegetation.vat

| Value | Count | Desc | Suit |
|---|---|---|---|
| 101 | 2450 | Grass | 1 |
| 201 | 65780 | Mixed | 3 |
| 301 | 32187 | Pine | 2 |
| 401 | 5433 | Oak | 5 |

Soil.vat

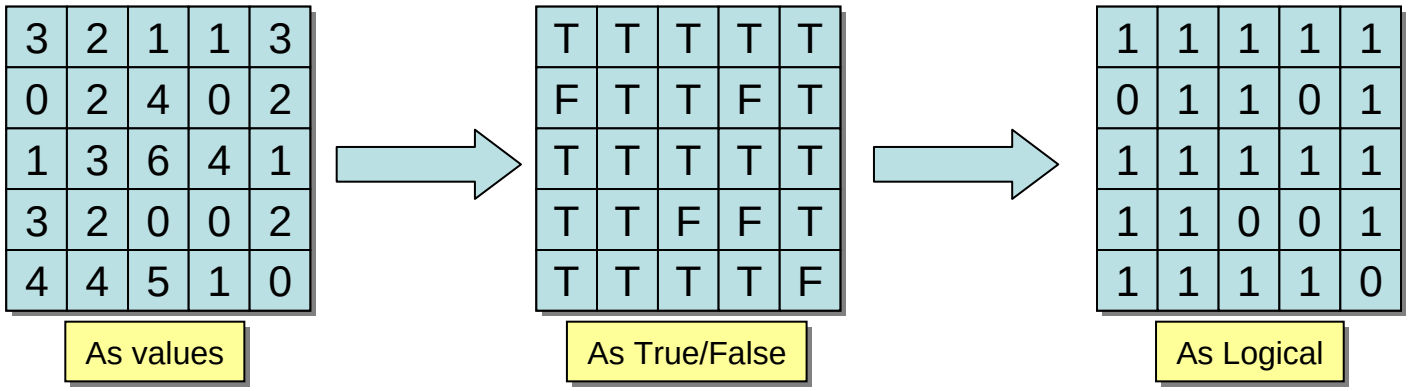
| Value | Count | Desc | Suit |
|---|---|---|---|
| 23 | 2450 | Sand | 2 |
| 46 | 65780 | Loam | 1 |
| 87 | 32187 | Clay | 6 |
| 99 | 5433 | Rock | 9 |

Vegetation.suit + soil.suit

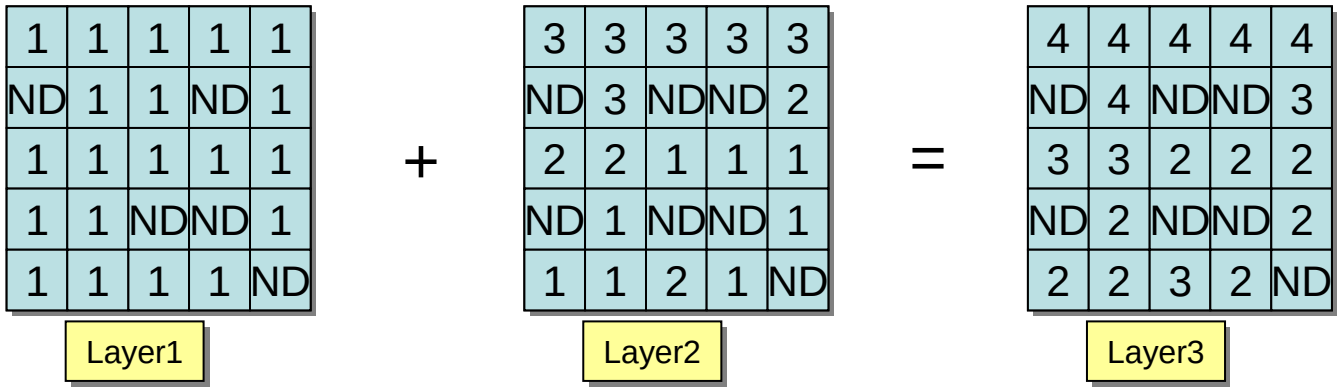
  - *Layer alone* is assumed to be Layer.Value
- You may join tables to raster VAT file
  - Use joined fields for symbology, selection
  - Cannot use in Map Algebra expressions

# Special cell values in Map Algebra

- **Logical - Non-zero values are *True*, *zero* is *False***




- **NoData - If any input is *NoData*, the output is *NoData***



# Map Algebra operators

- **Work with two inputs, like Slope GE 10**

| Boolean | |
|---|---|
| ^, NOT | Logical complement |
| &, AND | Logical And |
| I, OR | Logical Or |
| !, XOR | Logical Xor |

| Relational | |
|---|---|
| = =, EQ | Equal |
| ^=, <>, NE | Not Equal |
| <, LT | Less than |
| <=, LE | Less than or equal |
| >, GT | Greater than |
| >=, GE | Greater than or equal |

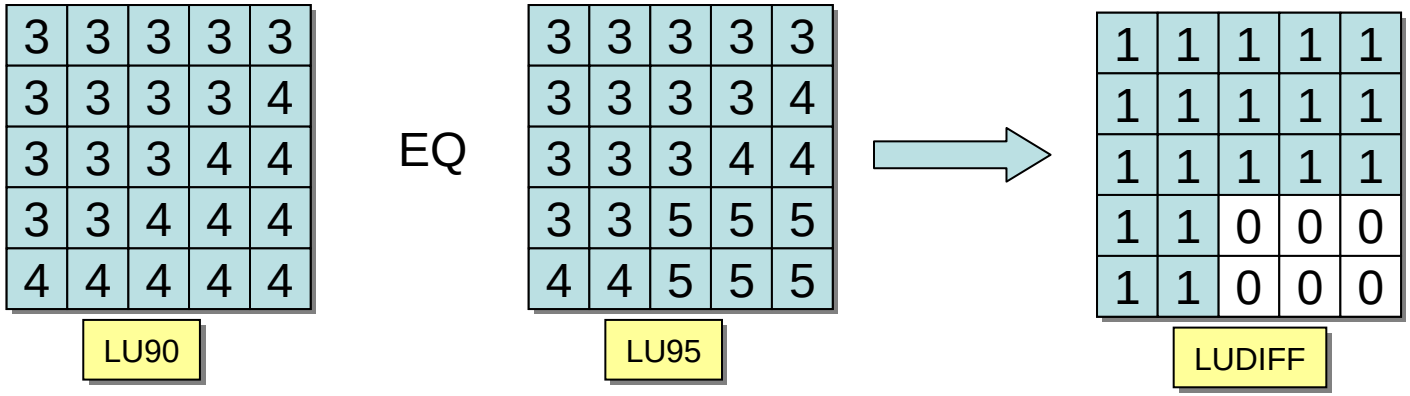
| Arithmetic | |
|---|---|
| + | Addition |
| - | Subtraction |
| * | Multiplication |
| /, DIV | Division |
| MOD | Modulus |
| - | Unary minus |

| Combinatorial | |
|---|---|
| COR | Combinatorial Or |
| CAND | Combinatorial And |
| CXOR | Combinatorial XOR |

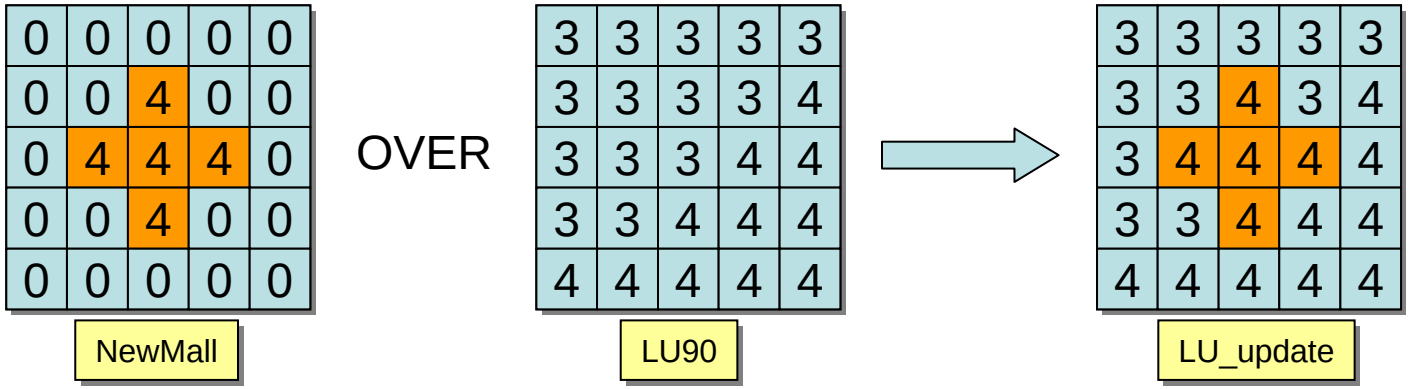
| Logical | |
|---|---|
| DIFF | Logical difference |
| IN {list} | Contained in list |
| OVER | Replace |

# Examples of operators

- **Using EQ to find changes in lad use**

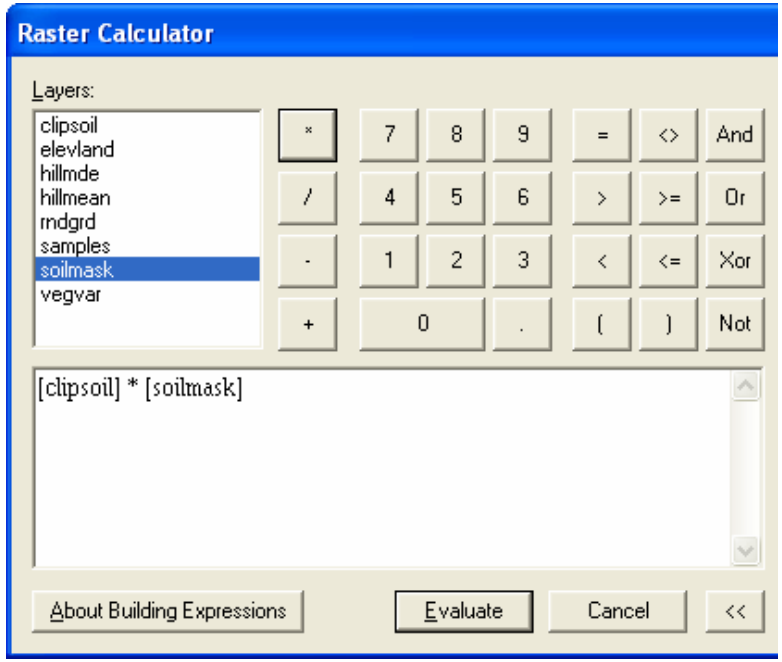



- **Using over to update land use**



# ArcGIS Spatial Analyst command

- There are a few commands
  - Tasks not suited to functions
  - Enter in Raster Calculator
- Not part of Map Algebra
  - Cannot use in expressions
- Mostly data management
  - Build attribute tables
  - Create, manage grid stacks
  - Others



# Exercise 4A overview

- Work with Map Algebra
- Operators
- Objects
- Commands

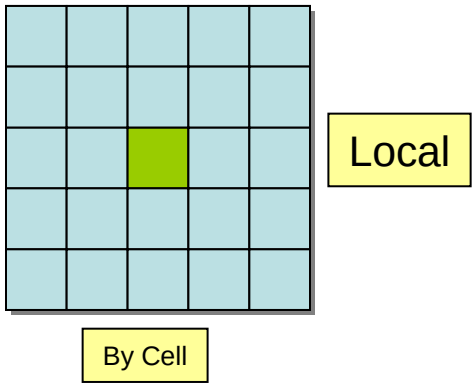
# Lesson 4B overview

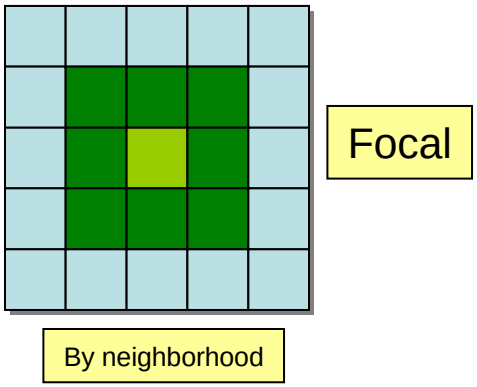
- Introduction to Map Algebra (4A)
  - Language components
  - Syntax and rules
  - Objects
  - Operators
  - Commands
  - Exercise 4A

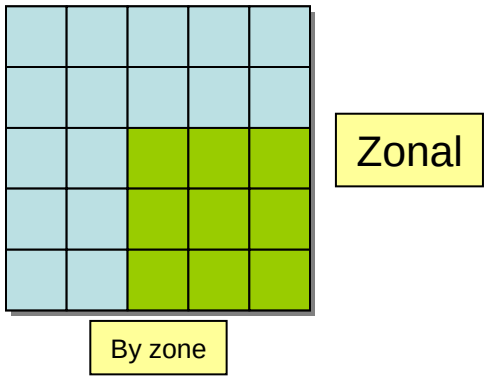
- Map Algebra functions (4B)
  - Function syntax
  - Local functions
  - Focal and Block functions
  - Zonal functions
  - Global functions
  - Exercise 5B

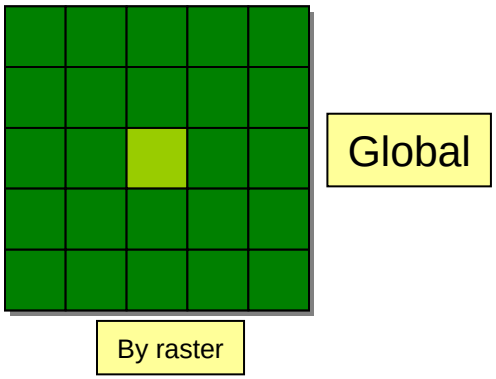
# Map Algebra functions

- Do most of the work in Map Algebra
  - About 168 functions
- Can classify functions by processing method



# Function syntax rules

- Functions return values
- Use as “objects” in expressions

```
Farmsoils + CON(S1ope LE 15, 1, 0)
```

- Arguments in parentheses and comma-delimited

```
SLOPE(Elevation, DEGRE)
```

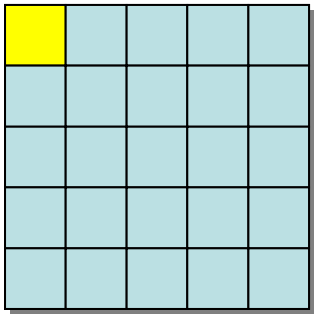
- Arguments may be other functions or expressions*

```
SLOPE (IDW(c:\data\elevpoints, spot), DEGREE)
```

- * If they return the data expected by the argument
- * If the function allows it

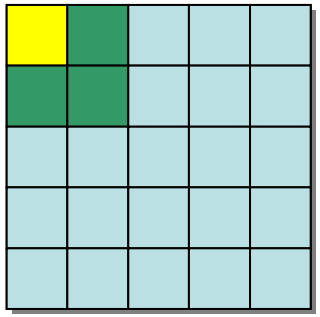
# Local functions

- Compute values based on the current output cell
  - Most functions are local
- Each cell is processed, starting at top left



# Focal functions

- Compute values by cell neighborhoods
  - Writes result to current cell in the output grid
- Neighborhood is a moving window over input

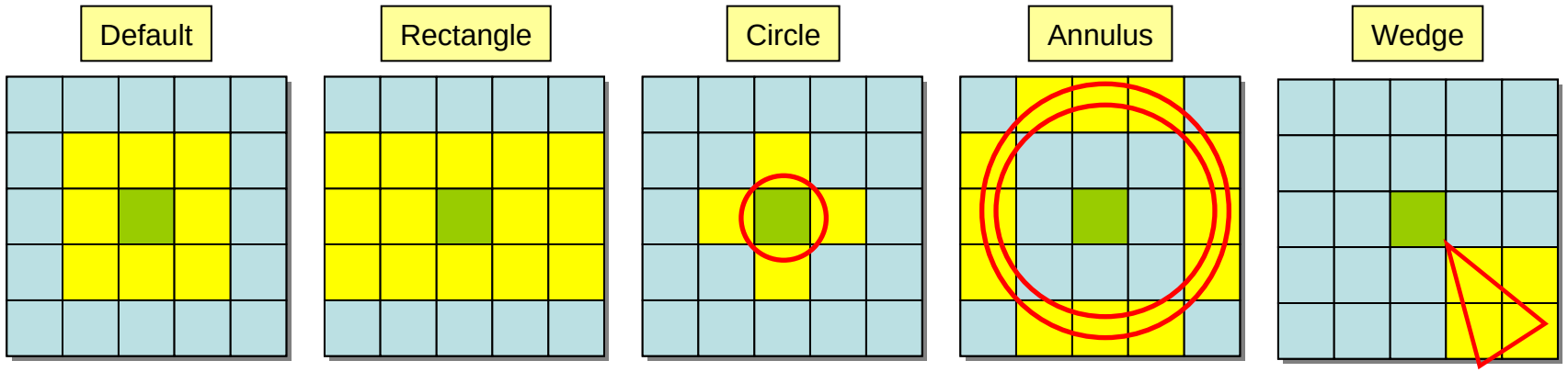


- General syntax:

```
FOCALxxx(InLayer, neighborhood, (DATA | NODATA))
```

# Focal neighborhoods

- You may define the neighborhood geometry




```
FOCALSUM (Elev)
FOCALSUM (Elev, Rectangle, 5, 3)
FOCALSUM (Elev, Circle, 2)
FOCALSUM (Elev, Annulus, 2, 3)
FOCALSUM (Elev, wedge, 4, 300, 330)
```

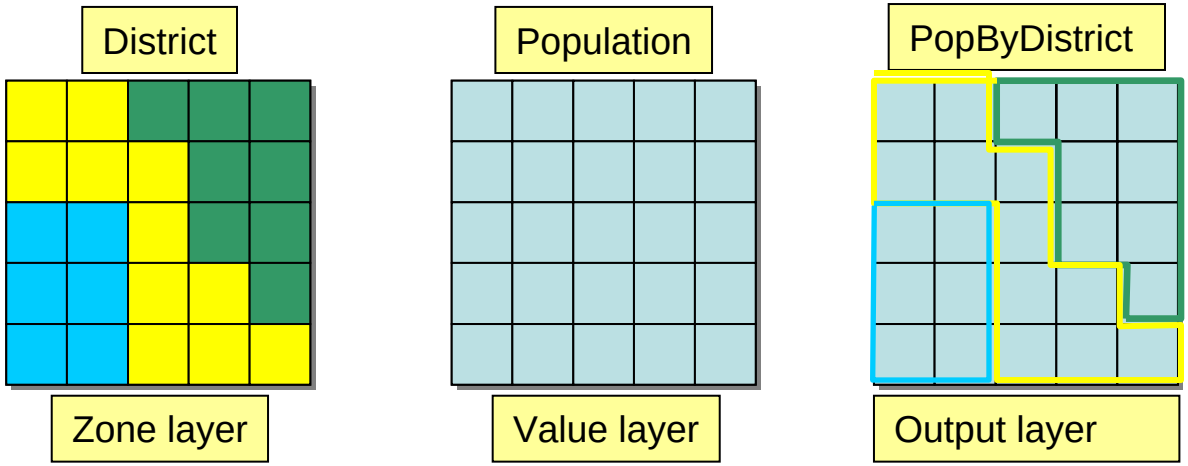
Kernel.txt

```
5 3
1 1 0 1 0
1 0 0 1 1
0 1 1 1 0
```

- Create a custom neighborhood with a kernel file

# Zonal functions

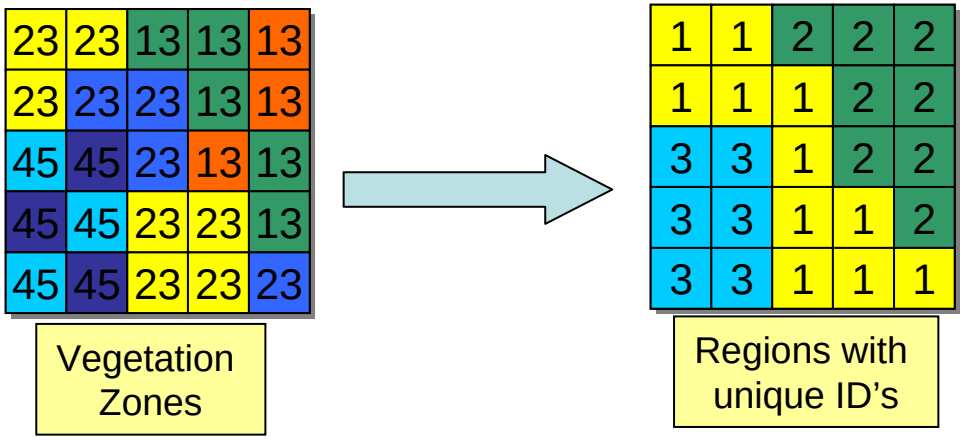
- **Most summarize values in a layer by zones in another**
  - **Require two input rasters: Zone and Value**




ZonalSum(District, Population)

# Global functions

- **May access all input cells to compute output cell value**
  - **Mostly do distance**
- **REGIONGROUP assigns unique ID's**



# The CON function

- **IF-THEN-ELSE function for Map Algebra**

```
CON(<condition>, <true_expression>, {false_expression})
IF    TEST          THEN(TEST= 1)       ELSE(TEST=0)
```

- **May be a simple IF-THEN-ELSE**

```
CON(Slope <20, 1, 0)
```

- **Or nest CONs for an ELSE-IF**

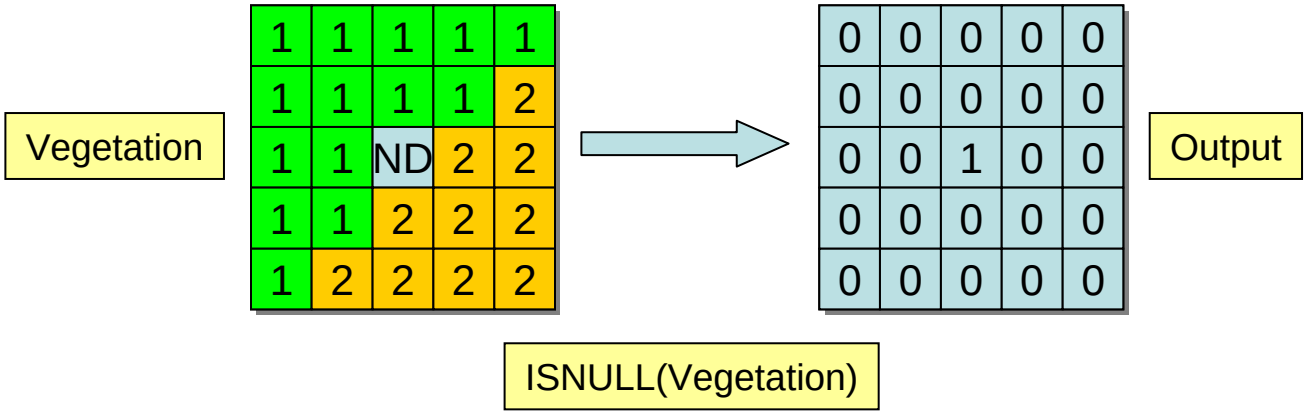
```
CON(Slope < 20, 1,
    CON(Slope < 40, 2,
        CON(Slope < 90, 3, 5) ) )
```

First CON {false expression}

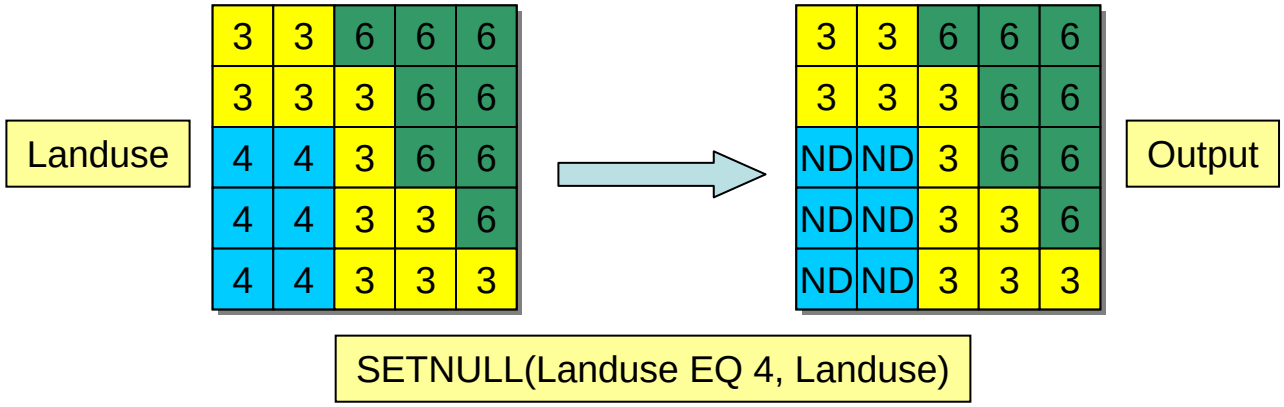
Second CON {false_expression}

# Working with NoData

- **ISNULL *tests* for NoData: Returns *true or false***




- **SETNULL assiqns NoData to cells that pass a test**



# Exercise 4B overview

- Work with Map Algebra functions
- Work with local functions
- Work with focal functions
- Work with zonal functions
- Work with global functions