CS 221 Lecture

Tuesday, 13 September 2011

Today's Agenda

1. Announcements
2. Boolean Expressions and logic
3. MATLAB Fundamentals

1. Announcements

- First in-class quiz: Tuesday 4 October
- Lab quiz: Thursday 29 September
  - Excel fundamentals
  - Excel conditionals
  - Basic MATLAB computations
- Homework Assignment 1 available on web site
  - Due next Wednesday, 21 September, 11:59 pm
  - Grading rubric available by Thursday
- Lab 2 is due Thursday
- Remember:

  Bring your textbook to lab!

2. Boolean Expressions and Logic

Learning Arithmetic & Algebra

Remember when you learned...

Whole number arithmetic:

- whole numbers (0, 1, 2, ...)
- Operators: addition (+); subtraction (−); multiplication (×); division (÷)

addition/subtraction facts (5+7=12, 9-4=5)

big numbers, carrying/borrowing

multiplication tables

multiplying multi-digit numbers

long division (with remainders)

fractions

Learning Arithmetic & Algebra

Integer Arithmetic:

- integers (..., -2, -1, 0, 1, 2, ...)
- Operators: addition (+); subtraction (–); multiplication (×); division (÷)

addition/subtraction facts;

multiplying/dividing positive/negative numbers

Learning Real Algebra

Real Arithmetic:

- Real numbers (0, 1, 2, ...)
- Operators ...
- Laws/properties:

  Commutative: $a + b = b + a$, $a \times b = b \times a$

  Associative: $(a + b) + c = a + (b + c)$

  $(a \times b) \times c = a \times (b \times c)$

  Distributive: multiplication distributes over addition

  $a \times (b + c) = (a \times b) + (a \times c)$

  Identity (Unit): $0 + a = a$, $a - 0 = a$

  $1 \times a = a$, $a \div 1 = a$

  Zero: $0 \times a = 0$

Boolean Algebra

- Only two values: true and false
- Operators:
  - unary negation: NOT (Excel) ~ (MATLAB)
    - ~false = true, ~true = false
  - conjunction $\wedge$: AND (Excel) &&, & (MATLAB)
    - false $\wedge$ false = false
    - false $\wedge$ true = false
    - true $\wedge$ false = false
    - true $\wedge$ true = true
  - disjunction $\vee$: OR (Excel) ||, | (MATLAB)
    - false $\vee$ false = false
    - false $\vee$ true = true
    - true $\vee$ false = true
    - true $\vee$ true = true

Boolean Algebra

- Laws/Properties:

  Disjunction and Conjunction are Associative and Commutative:

  $$a \wedge b = b \wedge a, \quad a \vee b = b \vee a$$

  $$(a \wedge b) \wedge c = a \wedge (b \wedge c)$$

  $$(a \vee b) \vee c = a \vee (b \vee c)$$

  Distributive: AND, OR distribute over each other

  $$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$$

  $$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$$

  Identity (Unit): $\text{true} \wedge a = a$, $\text{false} \vee a = a$

  Zero: $\text{false} \wedge a = \text{false}$, $\text{true} \vee a = \text{true}$

Conditionals in Programming

- Both Excel and MATLAB use Boolean Expressions to define conditionals:
  - IF(<boolean expr>,<value if true>,<value if false>)

    if first argument evaluates to true, value is 2nd arg, else (evaluates to false) 3rd arg.
  - if <boolean expr> <statement> end

    if boolean expr evaluates to true, execute the statement, else (evaluates to false) skip it.
- Conditionals allow the result of the computation to vary with the data

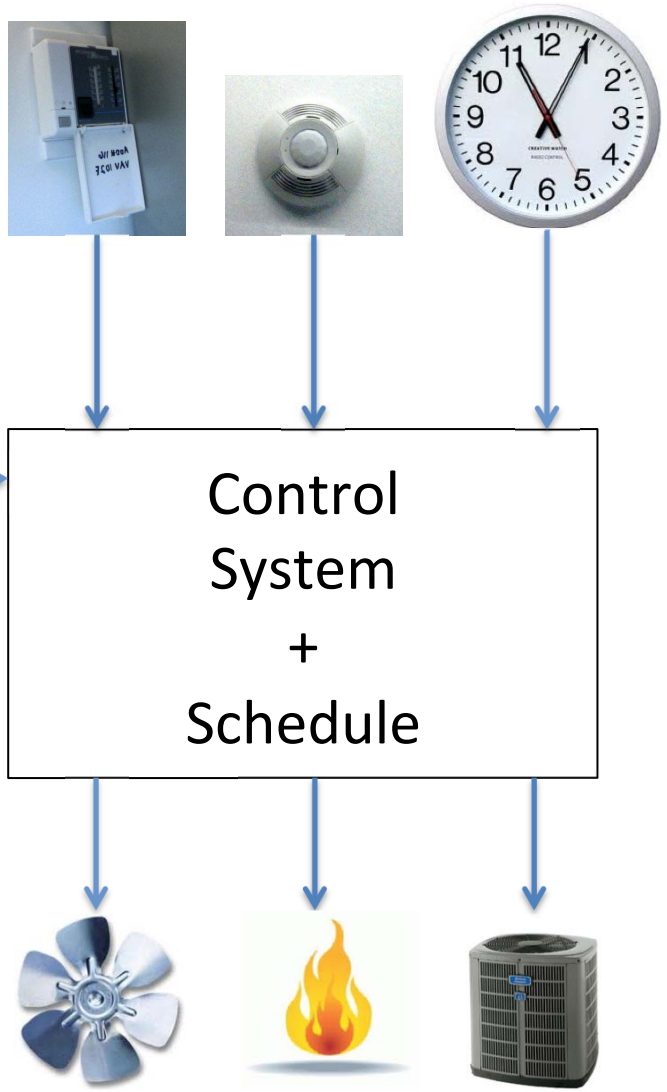
An Example: HVAC Control

- The "Delta Room" oversees heating and cooling operations of most buildings on campus
- Sensors in buildings report temperature, thermostat settings, and other information
- Equipment is controlled via outputs from the system

photo by Brandon Goodwin
© Kentucky Kernel

Example: HVAC Control of an Office

- Control System Inputs:
  - temperature sensor (degrees F)
  - thermostat setting (degrees F)
  - occupancy sensor (boolean)
    - true = occupied
    - false = unoccupied
  - schedule override (boolean)
    - true = override is in effect
    - false = no override
  - current time
- Control System Outputs:
  - fan control ("ON"/"OFF")
  - heating control ("ON"/"OFF")
  - cooling control ("ON"/"OFF")



Example: HVAC Control of an Office

- Schedule:
  - Normal operating hours are 7 a.m. to 11 p.m.
- Policy:
  - Cooling unit is ON whenever the temperature is above the thermostat setting, the office is occupied, and the current time is during normal operating hours or the schedule override is on; otherwise OFF.
  - Heating unit is ON whenever the temperature is below the thermostat setting, the office is occupied, and the current time is during normal operating hours or the schedule override is on; otherwise OFF.
  - Fan is ON whenever heating or cooling unit is on.

Modeling the Logic

- Variable/cell for each input and output
- Values of output variables depend on input variables
  - Use assignment: output = f(input)
- Cooling unit should be on if:

X

"temperature is above the thermostat setting, the office is occupied, and

the current time is during normal operating hours or

the schedule override is on" Z

Y

  - This has the form: "X and Y or Z"

    (X and Y) or Z vs. X and (Y or Z)

    ...Does it matter?

Truth Tables

- A way to compute the value of a boolean expression
- A matrix with:
  - One column for each boolean subexpression, plus one for the whole expression
  - One row for each possible combination of subexpression values
- Examples:

X	Y	X∧Y
F	F	F
F	T	F
T	F	F
T	T	T

X	Y	X∨Y
F	F	F
F	T	T
T	F	T
T	T	T

Parsing the Specification

3 variables, so $2^3 = 8$ rows

X	Y	Z	X∧Y	Y∨Z	(X∧Y)∨Z	X∧(Y∨Z)
F	F	F	F	F	F	F
F	F	T	F	T	T	F
F	T	F	F	T	F	F
F	T	T	F	T	T	F
T	F	F	F	F	F	F
T	F	T	F	T	T	T
T	T	F	T	T	T	T
T	T	T	T	T	T	T

Yes, it matters. The specification is ambiguous!
We interpret it as X ∧ (Y ∨ Z) – "Override" applies to the schedule only.

Parsing the Specification

- Yes, it matters!
- The specification is ambiguous
  - Alas, this is not unusual with natural-language specifications
- We have to look for clues to the correct interpretation
  - "Schedule override" suggests that the override only applies to the schedule
  - So the expression should be false whenever the first condition (temp out of range and occupied) is false
  - The correct interpretation is $X \wedge (Y \vee Z)$

Breaking It Down

- Cooling control output:
  - "Temperature above thermostat setting"
    temperature > thermostat
  - "Office is occupied"
    occupied
    - **Note**: never "occupied equals true"
    - For any boolean expression x,
      "x equals true" is exactly the same as "x"
  - "Current time is during normal operating hours"
    = "Current time is between normal ops start and end"
    Current time > Normal_start AND
    Current time < Normal_end
  - "Override is on"
    override

Computing with Times (and Dates)

- How to represent dates and times in the computer?
  - Typically "time" means "date and time"
- Possibilities:
  - A 5-tuple of values

    [year, month, day, hour, minute, second]
  - A string

    "9/14/11" or "14 September 2011" or "2011.09.14"
  - Number of days (and/or fractions of days, i.e., time) since some epoch
    - Both Excel and MATLAB use this method
    - Both have helpful functions to convert between forms

    Note: it doesn't much matter what the epoch is, as long as it is reasonably far in the past

Getting the Current Time/Date

- Excel: function "NOW()"
  - returns number of days since Jan 1, 1900
  - A double-precision number
  - Fraction represents the current time
- MATLAB: special variable "now"
  - number of days since Jan 1, 0000
  - A double-precision number
- One second = 1/86400 day

  = 0.000011574 day

  NOTE: the value returned is nowhere near as accurate as its precision would suggest

Comparing Times

- Excel:
  - HOUR(NOW()) returns the current hour
    - an integer
  - Compare to starting hour and ending hour:

    AND(HOUR(NOW()) > 7 [start of interval],
    HOUR(NOW()) < 23 [end of interval])

    has to be true
- MATLAB:
  - Use datevec() to convert serial to vector of values

    [year month date hour min sec]
  - Select 4th element, compare to start/end hours

Putting It Together

- Cooling unit control:

```
currenthour:   =HOUR(NOW())
duringHours:   =AND(currenthour >= startHour,
                    currenthour < endHour)
=IF(AND(temp>setting,
        AND(occupied,
            OR(duringHours,override))),
    "ON",
    "OFF")
```

- Heating unit control similar
- Fan on if either heating or cooling is on

3. MATLAB Funda’s

Using MATLAB

- As a calculator
  - Example: calculate the volume of a cylinder

    Volume = $\pi r^2 h$

MATLAB Script (".m") Files

- You can save a sequence of commands in a file for MATLAB to "play back" any time
- These are called script files
  - E.g., cylvol.m
- Invoke by typing the name of the file (without the .m)
  - MATLAB executes the lines in the file one by one, exactly as if you had typed them
  - **Note**: variables appear in the workspace
  - **Note**: to suppress output, use semicolon;

What Happens When You Type Something at the Command Line

- MATLAB first parses what you typed, to determine what kind of thing it is:
  - Expression
    - Assignment expression ("var = expr...")
      - Evaluate the part on the RHS and change the value of variable on the LHS to the result (print if no ;)
    - Other expression (no assignment)
      - Evaluate the expression and change the value of "ans" to the result (print if no ;)
      - Note: a variable name is an expression
  - Word (no operators): try each of these in turn
    - Variable name – treat as expression (see above)
    - Built-in command: clear, format, clc, etc.
    - Name of a .m file

Path: Where MATLAB Looks for Scripts

- Path: a list of directories (folders) where MATLAB searches for files with names matching commands you typed
- Initially includes many MATLAB directories
- May or may not contain your current directory
  - MATLAB warns if it is not in your path
- You can modify the Path:
  - Via the Current Directory Window
  - Via command-line: see "help path"
  - Be Careful about removing folders – you may have to restart MATLAB to get them back

Conditionals in MATLAB

- In Excel, function IF(boolexp,tvalue,fvalue) returns a value that depends on the value of boolexp
- In MATLAB, the basic "if" controls whether a command is executed:

```
if <boolexp>
    <command>   % executed iff <boolexp> is true
end
```

If <boolexp> evaluates to false (zero), the if does nothing!

Other forms of if-statements

```
if  <boolexp>
     <command1>
else
     <command2>
end
```

<boolexp> evaluates to true (nonzero):
    execute <command1>

<boolexp> evaluates to false (zero):
    execute <command2>

Using if-statements

Sometimes you need to test a bunch of conditions:

```
if  score >= 90
     grade = 'A';
else
     if score >= 80
          grade = 'B';
     else
          if score >= 70
               grade = 'C';
          else
               grade = 'E';
          end
     end
end
```

Using if-statements

The "elseif" form of if-statement just makes this cleaner:

```
if  score >= 90
    grade = 'A';
elseif score >= 80
    grade = 'B';
elseif score >= 70
    grade = 'C';
else
    grade = 'E';
end
```

- Only one "end" is required
- Less indentation

A Note About Formatting

- ALWAYS format your code properly
  - Statements/commands inside an "if" (or other compound statements) should be indented:

```
if  temp > setting
     control = 'on';
end
```

  - Be consistent about the amount you indent
  - A few spaces (2-4) is best
    - Tabs make lines too long
- Why this is important:
  - Program text is the (only!) carrier of our understanding
  - Use the built-in editor – it will (usually) do the right thing

Example: Printing Info About a Number

- Get a number from the user.
- Say whether it is negative, zero or positive.
- Say whether it is an integer.
- If it is an integer, say whether it is divisible by 2, 3, or 5.
- Use the “disp()” function to produce output.

Summary

Take-aways: Boolean Logic

- Boolean expressions are important for capturing the logic of a problem or situation
  - Boolean algebra is like algebra of real numbers
- Understanding the logic of a problem requires breaking it down into pieces
  - Variables (input, output, intermediate...)
  - Relationships among variables
  - Conditions involving variables
- Problem specifications in English may be ambiguous
- Truth tables can be handy for understanding complicated expressions

Take-aways: MATLAB

- Scripts (m-files) allow you to re-use computations
  - input() function allows interaction with the user
- Conditionals control the execution of statements
- 3 forms:
  - if <boolexp> <command> end
  - if <boolexp> <command1> else <command2> end
  - if <boolexp1> command1> elseif <boolexp2> <command2> else <command3> end
- Indentation is very important for understanding code