Boolean algebra, conditional statements, loops.

Eugeni E. Mikhailov

The College of William & Mary

Lecture 04
Boolean algebra

Variable of valuable type can have only two values
- true (Matlab use 1 to indicate it, actually everything but zero)
- false (Matlab uses 0)

There are three logical operators which are used in boolean algebra
- ¬ - logic not, Matlab ~
  \[ \neg \text{true} = \text{false} \]
  \[ \neg \text{false} = \text{true} \]
- ∧ - logic and, Matlab &
  \[ A \land B = \begin{cases} 
  \text{true, if } A=\text{true and } B=\text{true}, \\
  \text{false, otherwise} 
  \end{cases} \]
- ∨ - logic or, Matlab |
  \[ A \lor B = \begin{cases} 
  \text{false, if } A=\text{false and } B=\text{false}, \\
  \text{true, otherwise} 
  \end{cases} \]
Boolean operators precedence in Matlab

If $A = false$, $B = true$, $C = true$

$$A \mid \sim B \& C$$

$\sim$ has highest precedence, then $\&$, and then $\mid$

$$A \mid ((\sim B) \& C)$$

Thus

$$A \mid \sim B \& C = false$$

“Cat is an animal and cat is not an animal” is false statement

$$\sim Z \& Z = false$$
There is an island, which is populated by two kind of people: liars and truthlovers.

- Liars always lie and never speak a word of truth.
- Truthlovers always speak only truth.

Suppose, you are landed on this island and met a person. What will be the answer to your question “Who are you?”

- The answer always will be “Truthlover”.

Now you see a person who answers to your question. “I am a liar.” Is it possible?

- This makes a paradox and should not ever happen on this island.
Matlab boolean logic examples

- $123.3 \ & \ 12 = 1$
- $\sim 1232e-6 = 0$

```matlab
>> B=[1.22312, 0; 34.343, 12]
B =
1.2231    0
34.3430   12.0000
```

```
~B
ans =
0     1
0     0
```

```
B \ | \ ~B
```

```
ans =
1     1
1     1
```

“To be or not to be”
Matlab boolean logic examples

>> B=[1.22312, 0; 34.343, 12]
B =
1.2231 0
34.3430 12.0000

>> A=[56, 655; 0, 24.4]
A =
56.0000 655.0000
0 24.4000

B&A
ans =
1 0
0 1

A|~B
ans =
1 1
0 1
## Comparison operators

<table>
<thead>
<tr>
<th>Math</th>
<th>Matlab</th>
</tr>
</thead>
<tbody>
<tr>
<td>(=)</td>
<td>(==) double equal sign!</td>
</tr>
<tr>
<td>(\neq)</td>
<td>(\sim=)</td>
</tr>
<tr>
<td>(&lt;)</td>
<td>(&lt;)</td>
</tr>
<tr>
<td>(\leq)</td>
<td>(\leq)</td>
</tr>
<tr>
<td>(&gt;)</td>
<td>(&gt;)</td>
</tr>
<tr>
<td>(\geq)</td>
<td>(\geq)</td>
</tr>
</tbody>
</table>

```matlab
x=[1,2,3,4,5]
x =
    1    2    3    4    5
x >= 3
ans =
    0    0    1    1    1
% chose such 'x' where x>=3
x(x >= 3)
ans =
    3    4    5```
Comparison with matrices

>> A = [1, 2; 3, 4]
A =
1   2
3   4

>> B = [33, 11; 53, 42]
B =
22  11
53  42

A >= 2
ans =
0   1
1   1

A (A >= 2)
ans =
3
2
4

B (A >= 2)
ans =
53
11
42
if expression
this part is executed
only if expression is true
else
this part is executed
only if expression is false
end

if hungry
buy some food
else
keep working
end

if (x >= 0)
    y = sqrt(x);
else
    error('cannot do');
end
Common mistake in the ’if’ statement

```c
if (x==y)
    D=4;
    Z=45;
    C=12;
else
    D=2;
end
```

the value of ’D’ is always 4, except the case when y=0
someone used assignment operator (=) instead of comparison (==)
if expression
this part is executed only if expression is true
end

if won a million
  go party
end

if (deviation<=0)
  exit;
end
The 'while' statement

```plaintext
while expression
this part is executed while expression is true
end
```

```plaintext
i=1;
while (i<=10)
c=a+b;
z=c*4+5;
i=i+2;
end
```

while loop is extremely useful but they are not guaranteed to finish. For a bit more complicated conditional statement and loop it is impossible to predict if the loop will finish.

Yet another common mistake is

```plaintext
i=1;
while (i<=10)
c=a+b;
end
```

not updating the term leading to fulfillment of the while condition
The 'for' statement

for variable = expression
do something
end

In this case variable is assigned consequently with columns of the expression, and then statements inside of the loop are executed

for loops are guaranteed to complete after predictable number of iterations (the amount of columns in expression).

```
sum=0;
x=[1,3,5,6]
for v=x
    sum=sum+v;
end

>> sum
sum =
    15
```
Example

\[ S = \sum_{i=1}^{100} i = 1 + 2 + 3 + 4 + \cdots + 99 + 100 \]

\begin{align*}
S &= 0; \ i = 1; \\
\text{while} \ (i \leq 100) \\
&\quad S = S + i; \\
&\quad i = i + 1; \\
\text{end}
\end{align*}

\begin{align*}
S &= 0; \\
\text{for} \ i = 1:100 \\
&\quad S = S + i; \\
&\quad S = S + i; \\
\text{end}
\end{align*}
Example

\[ S = \sum_{k=1}^{\infty} a_k \]

Until \( k \leq 100 \) and \( a_k < 10^{-5} \), where \( a_k = k^{-k} \).

\[
S=0; \quad k=1;
\text{while}( \ (k\leq100) \ & \ (k^\text{-}k \geq 1\text{e}-5) \ )
  \quad S=S+k^\text{-}k;
  \quad k=k+1;
\text{end}
\]

\[
\gg \ S
S = 1.2913
\]

\[
S=0; \quad k=1;
\text{while}( \ k\leq100 \ )
  \quad a_k=k^\text{-}k;
  \quad \text{if} \ (a_k < 1\text{e}-5)
      \quad \text{break};
  \quad \text{end}
  \quad S=S+a_k;
  \quad k=k+1;
\text{end}
\]

\[
\gg \ S
S = 1.2913
\]
Same example with 'for' loop

\[ S = \sum_{k=1}^{\infty} a_k \]

Until \( k \leq 100 \) and \( a_k < 10^{-5} \), where \( a_k = k^{-k} \).

\[
\begin{align*}
S &= 0; \\
\text{for } k &= 1:100 \\
& \quad a_k = k^{-k}; \\
& \quad \text{if } (a_k < 1e-5) \\
& \quad \quad \text{break;} \\
& \quad \end{align*}
\]

\[
\begin{align*}
S &= S + a_k; \\
\text{end}
\end{align*}
\]

\[
\begin{align*}
\text{>> } S \\
S &= 1.2913
\end{align*}
\]
Interest rate related example

Suppose bank gave you 50% interest rate (let’s call it ’x’), and you put one dollar in. How much would you get at the end of the year?

- one payment at the end of the year

\[ M_1 = 1 \times (1 + x) = 1 \times (1 + .5) = 1.5 \]

- interest payment every half a year

\[ M_2 = 1 \times (1 + x/2) \times (1 + x/2) = 1 \times (1 + .5/2)^2 = 1.5625 \]

- interest payment every month

\[ M_{12} = 1 \times (1 + x/12)^{12} = 1.6321 \]
Now let’s find how your money growth \( M_N \) depends on the number of payments per year.

```matlab
N_max=100;
N=1:N_max;
M=0*(N);
x=.5;
for i=N
    M(i)=(1+x/i)^i;
end
plot(N,M,’-’);
xlabel(’N, number of payments per year’);
ylabel(’Money grows’);
title(’Money grows vs number of payments per year’);
```
Interest rate related example

Money grows vs number of payments per year

Money grows

N, number of payments per year

Eugeni Mikhailov (W&M)  Practical Computing  Lecture 04