CS 160: Practice Sheet 4 (do not turn in)

#1 Attributed Grammars (Roman Numerals)

In case you have forgotten how roman numerals work, the following rules should refresh your memory:

i) Roman numerals are composed of the symbols I (one), V (five), X (ten), L (fifty), C (one hundred), D (five hundred), and M (one thousand).

ii) A smaller number placed in front of a larger number means you should subtract the smaller number from the larger number. Only one number can be subtracted (e.g., IIV does not mean 3, it is illegal)

iii) A number placed after a number with equal or greater value is added to it (e.g., DC means 600, XIII means 13, and MMM means 3000).

$$
\begin{aligned}
S &\rightarrow R \\
R &\rightarrow E\ R \\
&\mid E \\
E &\rightarrow \mathbf{M} \\
&\mid \mathbf{D} \\
&\mid \mathbf{C} \\
&\mid \mathbf{L} \\
&\mid \mathbf{X} \\
&\mid \mathbf{V} \\
&\mid \mathbf{I}
\end{aligned}
$$

Assuming that only valid roman numerals are passed in (i.e., you do not need to enforce that the numbers are valid, and your grammar can produce arbitrary output for invalid inputs), write an synthesized attribute grammar that will calculate the value (*.val*) for each node in the parse tree. (S.*val* should be 600 for the sentence "DC"). Note that you can (and likely will need to) introduce additional attributes (in addition to *val*).

CS 160: Practice Sheet 4: Solution

S	$\rightarrow$	R	[S.val = R.val;]

R_0	$\rightarrow$	$E\ R_1$	[
			if (E.val < R_1.last)
			{ R_0.val = R_1.val – E.val; }
			else
			{ R_0.val = R_1.val + E.val; }
			R_0.last = E.val;
			]

	|	E	[R_0.val = E.val; R_0.last = E.val;]
E	$\rightarrow$	**M**	[E.val = 1000;]
	|	**D**	[E.val = 500;]
	|	**C**	[E.val = 100;]
	|	**L**	[E.val = 50;]
	|	**X**	[E.val = 10;]
	|	**V**	[E.val = 5;]
	|	**I**	[E.val = 1;]