On the Matta Programming Language

Dedicated to Harsha and his strive to innovate

Lavaan Mathanmohan
Y12 CSSOC member

The MPL allows logical statements to be written as mathematical expressions, featuring boolean operators, if/else statements, iterations and arrays represented by various mathematical constructs. The language is Turing complete, MPL programs - including a prime checker, can run on graphical calculators such as Desmos. It may be further possible that output of MPL programs can be approximated as summations in $O(1)$ time.

Basic mathematical operators such as the modulo, floor and ceiling functions have interesting logical properties. These properties are inherited from their definitions, which contains the if statement. In this text we will explore how this property can be exploited to create a Turing complete language.

DEFINING THE OPERATORS

The Floor function

The floor function, denoted as $\lfloor x \rfloor$, takes a real number as an input and returns the largest integer, which is less than the initial real number.

So, $\lfloor 3.7 \rfloor = 3$ and $\lfloor -3.7 \rfloor = -4$

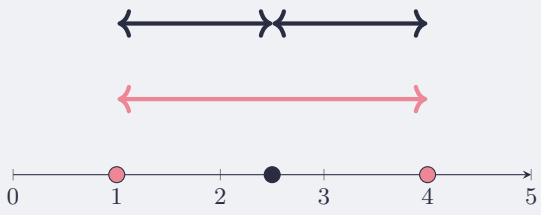
The Ceiling function

The ceiling function, denoted as $\lceil x \rceil$, takes a real number as an input and returns the smallest integer, which is larger than the initial real number.

So, $\lceil 3.7 \rceil = 4$ and $\lceil -3.7 \rceil = -3$

The Min and Max functions

I want the min function to take two real values as parameters and return the smaller of those real values. Then, for the max function I want to return the larger of those real values.



The two red points are the arguments of our function. We pick one of these points to be the value of the variable a. The other point is the value of variable b. Thus, we can say that the midpoint (the black spot) can be defined as $\frac{a+b}{2}$. As well as that, the distance between the two points can be defined as $|a-b|$.

Thus, we can also say that the distance between the midpoint and an input is $\frac{|a-b|}{2}$. If we move to the right by this distance from the midpoint, we will always get to the larger value and if we move to the left by this value from the midpoint, we will always get to the smaller value. Therefore we can say that

$$\min(a,b) \equiv \frac{a+b}{2} - \frac{|a-b|}{2}$$

and also

$$\max(a,b) \equiv \frac{a+b}{2} + \frac{|a-b|}{2}$$

The Not function

I want this function to either take 0 or 1 as an argument and return the boolean inverse of that value.

$$\begin{aligned} \text{not}(a) &\equiv 1 - a \equiv 0 \qquad \text{if } a \equiv 1 \\ \text{not}(a) &\equiv 1 - a \equiv 1 \qquad \text{if } a \equiv 0 \end{aligned}$$

This matches the truth table of the "not" gate, therefore we can say that

$$\text{not}(a) \equiv 1 - a$$

The Or and And functions

I want these functions to take an n number of 0s and 1s.

- And: if a single argument is 0, then it should return 0, otherwise it should return 1.
- Or: if a single argument is 1, then it should return 1, otherwise it should return 0.

If the average of all the arguments is below 1, then there was at least one argument which was 0. If it is 1 however, it means that all of the arguments were one. If we take the floor of the average, then if the average is below 1, then 0 will be returned, otherwise 1 is returned. This is the And function.

If we take the ceiling of the average, if the average is above 0, 1 is returned. This is the Or function. Therefore we can say that

$$\text{and}(s_1, s_2, ..., s_n) \equiv \left\lfloor \frac{\sum_{x=1}^{n} s_x}{n} \right\rfloor$$
$$\text{or}(s_1, s_2, ..., s_n) \equiv \left\lceil \frac{\sum_{x=1}^{n} s_x}{n} \right\rceil$$

The IsLesser function

I want this function to take two real arguments in a and b, in said order and return 1 if and only if the inequality $a < b$ is true, and return 0 otherwise.

$$\frac{b-a}{|a-b|} \equiv 1 \qquad \text{if and only if } b > a$$
$$\frac{b-a}{|a-b|} \equiv -1 \qquad \text{if and only if } a > b$$

If we add 1 to the denominator of this expression, we get $\frac{b-a}{|a-b|+1}$. Even as $|a-b|$ becomes infinitely large, $-1 < \frac{b-a}{|a-b|+1} < 1$, since the denominator will always be larger than the absolute value of the numerator.

We then use the logic $\lceil x \rceil \equiv 1$ and $\lceil -x \rceil \equiv 0$, where $0 < x < 1$. Thus $-1 < -x < 0$ to say that

$$\left\lceil \frac{b-a}{|a-b|+1} \right\rceil \equiv 0 \qquad \text{if } b < a \quad (\text{because } b - a < 0)$$
$$\left\lceil \frac{b-a}{|a-b|+1} \right\rceil \equiv 0 \qquad \text{if } a \equiv b$$
$$\left\lceil \frac{b-a}{|a-b|+1} \right\rceil \equiv 1 \qquad \text{if } b > a \quad (\text{because } b - a > 0)$$

Thus, $\left\lceil \frac{b-a}{|a-b|+1} \right\rceil \equiv 1$, if and only if $a < b$, otherwise $\left\lceil \frac{b-a}{|a-b|+1} \right\rceil \equiv 0$. Therefore we can say that

$$\text{isLesser}(a, b) \equiv \left\lceil \frac{b-a}{|a-b|+1} \right\rceil$$

The IsEqual function

I want this function to take two real values as arguments and return 1, if and only if those two values are equivalent, and otherwise 0 should be returned.

We know that $\lceil x \rceil \equiv 1$, where $0 < x < 1$.

We also know that for all for all real values of variables a and b, $0 \leq \frac{|a-b|}{|a-b|+1} < 1$. Since we are using the same logic from the proof of the isLarger function, but here the numerator is $|a - b|$, instead of $a - b$.

Thus, the we know that $\left\lceil \frac{|a-b|}{|a-b|+1} \right\rceil$ will always be 1, except from when $a \equiv b$, where it will be 0. Now if we invert these values using our Not function, where $\text{not}(a) \equiv 1 - a$, we can create the IsEqual function. Therefore we can say that

$$\text{isEqual}(a, b) \equiv 1 - \left\lceil \frac{|a-b|}{|a-b|+1} \right\rceil$$

The IsLesserOrEqual function

I want this function to return 1 if the inequality $a \leq b$ is true and otherwise return 0.

If we define set $A = \{x \mid x < K\}$, then set $B = \{x \mid x \geq K\}$ will contain only the values the set A does not contain and vice versa. Thus, $\text{not}(a \geq b)$ is the same as $a < b$, we can also say that $\text{not}(a \leq b)$ is the same as $b < a$, simply by inverting b and a.

Now we can substitute the MPL operators to re-write the previous sentence. By the way this programming language is referred to as MPL, the Matta programming language, or simply just Matta. Back to the point, we can say that

$$\text{not}(\text{isLesserOrEqual}(a, b)) \equiv \text{isLesser}(b, a)$$
$$\text{not}(\text{not}(\text{isLesserOrEqual}(a, b))) \equiv \text{not}(\text{isLesser}(b, a))$$
$$\text{isLesserOrEqual}(a, b) \equiv 1 - \text{isLesser}(b, a)$$
$$\text{isLesserOrEqual}(a, b) \equiv 1 - \left\lceil \frac{a-b}{|a-b|+1} \right\rceil$$

The floor division function

$$\text{div}(a, b) \equiv \left\lfloor \frac{a}{b} \right\rfloor$$

The floor division function, denoted as $\lfloor \frac{a}{b} \rfloor$, is simply the floor of a division. Thus, the floor division of two numbers is the quotient of the division of those two numbers

Arrays in MPL

This is simply a $1 \times n$ matrix and we can get an element at a certain index using this notation, (the first column of a matrix has an index of 1, instead of 0)

$$A \equiv \begin{pmatrix} 2 & 7 & 12 & 9 & 13 & 5 \end{pmatrix}$$
$$\text{elementAt}(A, \textit{index}) \equiv A_{\textit{index}+1}$$

The If-Elif-Else statement

I want to make a function which takes a $2 \times n$ real matrix as an argument, which will hold the values which will be returned if its corresponding logical statement is true.

There will of course be a priority in these logical statements, since that's how an *if elif else* statement works. The smaller the column number the value and the logical statement is under, the higher the priority they will have. Row 1 will hold the logical statements and row 2 will hold the returning values.

$$\text{ifElifElse}(A) \equiv \sum_{s=1}^{n} \left[A_{1,s} \times A_{2,s} \times \prod_{p=1}^{s-1} [\text{not}(A_{1,p})] \right] \qquad \text{where } A \in \mathbb{R}_{2\times n}$$

If we multiply a value by a logical statement, the result of the expression will be that value if that logical statement is true. The result of the expression will be 0 if the logical statement is false. The expression $57 \times \text{isEqual}(x, 7)$ will only be 57, if $x \equiv 7$, otherwise the expression is 0.

In this expression

$$(\textit{value}_1 \times \textit{statement}_1) + (\textit{value}_2 \times \textit{statement}_2 \times \text{not}(\textit{statement}_1))$$
$$+ (\textit{value}_3 \times \textit{statement}_3 \times \text{not}(\textit{statement}_1) \times \text{not}(\textit{statement}_2))$$

If $\textit{statement}_1$ is true, or in other words equivalent to 1, then all the other terms will be 0, since all the other terms are multiplied by $\text{not}(\textit{statement}_1)$. Since $\textit{statement}_1$ is true, the expression will be $\textit{value}_1$ in this scenario.

However, if $\textit{statement}_1$ is false and $\textit{statement}_2$ is true, then the first term and the last term will both be 0. Since in the first term $\textit{statement}_1 \equiv 0$ and in the last term $\text{not}(\textit{statement}_2) \equiv 0$, the expression will be $\textit{value}_2$ in this scenario.

Thus, we have prioritised logical statements and applied values to those logical statements. We can now take those logical statements and values as a logical argument and know which one must be prioritised, we can make a formula. This is why I have used a matrix.

If we say that

$$A \equiv \begin{pmatrix} \text{isLesser}(x,2) & \text{isLesser}(x,7) & \text{isLesser}(x,12) \\ 2 & 7 & 12 \end{pmatrix}$$

where the statement $\text{isLesser}(x,2)$ has the most priority and the statement $\text{isLesser}(x,12)$ has the least priority, then we can use the same logic as before, with formal notation.

As well as that, if the upper limit of the Pi notation is less than its lower limit, it will equal 1, so $\prod_{n=1}^{-5}[3] \equiv 1$.

Therefore we can say that

$$\text{ifElifElse}(A) \equiv \sum_{s=1}^{n}\left[A_{1,s} \times A_{2,s} \times \prod_{p=1}^{s-1}[\text{not}(A_{1,p})]\right] \qquad \text{where } A \in \mathbb{R}_{2\times n}$$

USING DESMOS AS AN IDE

Desmos is an online graphing calculator, which I have used to *run* programs I wrote in MPL.

An introduction to coding with Desmos

When writing $\lfloor x \rfloor$ or $\lceil x \rceil$, you have to write $\text{floor}(x)$ and $\text{ceil}(x)$ in Desmos. However, you can still write absolute values as $|x|$, in Desmos. To help myself, I made a Desmos account and defined all the operator functions and then saved that graph as *MPL Template*, so that I never have to rewrite that mess of ceiling functions and fractions.

One simple program to write could be $y = \text{isLesser}(4,x)$, which is $y = \left\lceil \frac{x-4}{|4-x|+1} \right\rceil$, where $y = 1$ if the inequality $4 < x$ is true, otherwise $y = 0$.

Using the IsEqual operator in Desmos

The IsEqual operator will return a different value for one singular point. This is not clearly shown if you graph it the usual way in Desmos. If you write $y = 1 - \left\lceil \frac{|x-7|}{|x-7|+1} \right\rceil$, you will not see a peak at $x = 7$, instead you will only see a straight line of $y = 0$.

However, you can get the integer plots in Desmos to see the peak at $x = 7$ clearly. By using this syntax, we can get integer plots:

$$[(X, 2X) \text{ for } X = [1, ..., 10000]]$$

10,000 is maximum number of elements in a list in Desmos, and in the above graph or list, we have used up 10000 of those elements. Now if we apply this syntax to our IsEqual operator, we can write

$$\left[\left(X, 1 - \text{ceil}\left(\frac{|X-7|}{|X-7|+1}\right)\right) \text{ for } X = [1, ..., 10000]\right]$$

MPL PROGRAMS

Digit of a natural number

This is the first program that was ever made in MPL.

$$\text{digit}(x, \Delta) \equiv \text{div}\left(x, 10^{\Delta-1}\right) - 10 \times \text{div}\left(x, 10^{\Delta}\right)$$

$$\{\text{digit}(x, \Delta) \mid x, \Delta \in \mathbb{N}\}$$

where x is the natural number and Δ is the Δ^{th} digit we are retrieving

Fizzbuzz

$$\text{fizzbuzz}(x, \Delta) \equiv \text{isEqual}(\text{mod}(x, 2\Delta + 3), 0)$$

$$\{\text{fizzbuzz}(x, \Delta) \mid x, \Delta \in \mathbb{Z}, 0 \leq \Delta \leq 1\}$$

where x is the number we are checking and Δ is mutates to check for a Fizz and a Buzz

- If $\Delta \equiv 0$, then $2\Delta + 3 \equiv 3$.
- If $\Delta \equiv 1$, then $2\Delta + 3 \equiv 5$

Thus when $\Delta \equiv 0$ we are checking for a Fizz, when $\Delta \equiv 1$ we are checking for a Buzz.

Prime checker

$$\text{isPrime}(x) \equiv \left\lfloor \frac{\sum_{n=2}^{x-1} \left(\lceil \frac{x}{n} \rceil - \lfloor \frac{x}{n} \rfloor\right)}{x-2} \right\rfloor$$

$$\{\text{isPrime}(x) \mid x \in \mathbb{N}, x > 2\}$$

The derivation for this function is an exercise for the reader.

SUMMARY

All the functions below, up to and including the or function only take 1 and 0 as arguments.

$$\begin{aligned}\text{not}(a) &\equiv 1-a\\ \text{and}(s_1,s_2,...,s_n) &\equiv \left\lfloor \frac{\sum_{x=1}^{n} s_x}{n} \right\rfloor\\ \text{or}(s_1,s_2,...,s_n) &\equiv \left\lceil \frac{\sum_{x=1}^{n} s_x}{n} \right\rceil\end{aligned}$$

All the functions below, up to and including the IsLesserOrEqual function only take real numbers as arguments.

$$\begin{aligned}\min(a,b) &\equiv \frac{a+b}{2} - \frac{|a-b|}{2}\\ \max(a,b) &\equiv \frac{a+b}{2} + \frac{|a-b|}{2}\end{aligned}$$

The floor division function is the quotient a division. $\text{div}(a,b) \equiv \lfloor \frac{a}{b} \rfloor$ The modulo function is the remainder of a division.

$$\text{mod}(a,b) \equiv a - \left\lfloor \frac{a}{b} \right\rfloor b$$

The IsLesser function takes the arguments,a and b, in said order and returns 1 if and only if the inequality $a < b$ is true and return 0 otherwise.

$$\begin{aligned}\text{isLesser}(a,b) &\equiv \ <(a,b)\\ &\equiv \left\lceil \frac{b-a}{|a-b|+1} \right\rceil\end{aligned}$$

The IsEqual function returns 1, if and only if those two arguments are equivalent, and otherwise 0 should be returned.

$$\begin{aligned}\text{isEqual}(a,b) &\equiv \ ==(a,b)\\ &\equiv 1 - \left\lceil \frac{|a-b|}{|a-b|+1} \right\rceil\end{aligned}$$

The IsLesserOrEqual function takes the arguments,a and b, in said order and returns 1 if the inequality $a \leq b$ is true and otherwise returns 0.

$$\begin{aligned}\text{isLesserOrEqual}(a,b) &\equiv \ <=(a,b)\\ &\equiv 1 - \text{isLesser}(b,a)\\ &\equiv 1 - \left\lceil \frac{a-b}{|a-b|+1} \right\rceil\end{aligned}$$

The ifElifElse function takes a $2 \times n$ real matrix as an argument, which will hold the values which will be returned if its corresponding logical statement is true. The further left the values and corresponding logical statements are positioned in the matrix, the higher their priority.

$$\text{ifElifElse}(A) \equiv \sum_{s=1}^{n} \left[A_{1,s} \times A_{2,s} \times \prod_{p=1}^{s-1} [\text{not}(A_{1,p})] \right] \qquad \text{where } A \in \mathbb{R}_{2\times n}$$