

0.0.1 Analysis of Algebraic Expressions

Algebraic Expressions are quite important in this project and as such it is important to break them down and understand what they really mean. The most basic idea in evaluating expressions is **BIDMAS**, which stands for **B**rackets, **I**ndices, **D**ivision, **M**ultiplication, **A**ddition and **S**ubtraction. It signifies the order that we must do operations on an expression (It is sometimes called **BODMAS** where the **O** stands for **O**rders) with the order of significance going from the first letter to the last. For example, let us use BIDMAS with the following expression:

$$3(4^2 + 2)$$

According to BIDMAS we will first look for brackets. There are brackets! The expression inside those brackets is $4^2 + 2$. We then apply BIDMAS again. There are no brackets but we do have indices. We apply the index function which produces $16 + 2$. Applying BIDMAS again we see that we must do addition which produces 18. Finally we have the expression $3(18)$ which is multiplication and it produces 54. Here we repeatedly took the most significant operator and applied its function until we fully evaluated our expression. Later we will instead take the least significant operator as this will allow us to split our expression down in a more structured manner.

In the example above there was nuance that we ignored. This was when we identified $3(18)$ actually meant $3 * (18)$. This implicitly represented multiplication and while as humans it is easy for us to process it, it is difficult for a computer to know this implicitly. It is therefore important that we remove these inconsistencies before we properly convert our input into a structure. It is also important that we strip away all whitespace before we start as this will allow our input to be more consistent. Here is a list of all these inconsistencies that we need to remove:

- Any instance of ax where $a \in \mathbb{R} : a \neq 0$ is to be converted to $a * x$.¹
- Any instance of $a($ and $)a$ where a is not an operator, is to be converted to $a * ($ and $)*a$ respectively..²
- Any instance of $(f(x))(g(x))$ is to be converted to $(f(x)) * (g(x))$.³
- Any instance of $! - f(x)$ where $!$ is to be any operator (e.g. $*$ or $/$) is to be converted to $!(-f(x))$.⁴
- Any instance of $-f(x)$ at the start or next to an opening bracket is to be converted to $0 - f(x)$.⁵

¹For example $4x$ is to be converted to $4 * x$

²For example $4(x + 1)$ is to be converted to $4 * (x + 1)$

³For example $(x + 4)(x - 3)$ is to be converted to $(x + 4) * (x - 3)$

⁴If there is a negate symbol next to another operator, we need to make sure that the negate symbol is not treated as an operator (even though we treat it like an operator in certain situations in the next step)

⁵Both these expressions are equivalent but the second allows us to reduce ambiguity if there is a negate symbol at the start of an expression e.g. $-x + 4$ would be treated as $0 - x + 4$ and $(-x)$ is $0 - x$.