

Formally describe the regular grammar of the lexemes in task 2.

$V = \{1, 2, 3, 4, 5, \dots, \text{Max Integer/Float}\}$

$S = \{+, -, *, /, p, d, i, ^\}$

$R = \{v_1 + v_2, v_1 - v_2, v_1 * v_2, v_1 / v_2 \text{ where } v_1 \geq v_2, p(v), d(v), i(v), ^{(v)}\}$

Describe the grammar of the infix notation in task 3 using (E)BNF. Beware of operator precedence. Is the grammar ambiguous? Explain why it is or is not ambiguous?

What is the difference between a context-sensitive and a context-free grammar?

The difference between a context-sensitive and context-free grammar is the way the rules work.

With context-free grammar (CFG), the rules are in the form of $v ::= y$ meaning where v is any variable from V , which is a finite set of variables that are non terminal and y any sequence of variables and symbols from $V \cup S$ where S is a finite set of symbols which are terminals.

In context-sensitive grammar (CSG), the rules are in a different form, the form of $\alpha v \beta ::= \alpha y \beta$, where v and y are the same as in context-free grammar, but $\alpha \beta$ are also any sequences from $V \cup S$

A clear difference is that CSGs are more complex than CFGs which needs more computational power, though CSGs can also have several terminals.

You may have gotten float-int errors in task 2. If you haven't, try running $1+1.0$. Why does this happen? Why is this a useful error?

When adding 1 and 1.0 together the error returned is says "Expected type: integer". Even though both are a type of number they are of different primitive types, Integers and Floats. When working with different data types it can be very helpful to be reminded that the types are not of the same origin. For example checking if $1 == 1.0$ will never become true, possibly making it harder to figure out where the fault lies. Oz is also a very simple language, were such operations should possibly not be considered.