1. A grammar for arithmetic expressions.

Arithmetic expressions are made from the following atomic elements ... with placeholder variables ...

- Var = variables x
- $\mathbb{R}$ = real numbers r
- $\mathbb{Z}$ = integers n
- Operators = $\{+,-,*,/\}$

A grammar for arithmetic expressions is

$$\begin{array}{rcl} AExp & a & \stackrel{\text{def}}{=} \quad a_1 + a_2 \quad | \quad a_1 - a_2 \quad | \quad t \\ term & t & \stackrel{\text{def}}{=} \quad t_1 * t_2 \quad | \quad t_1 / t_2 \quad | \quad f \\ factor & f & \stackrel{\text{def}}{=} \quad (a) \quad | \quad r \quad | \quad n \quad | \quad x \end{array}$$

(a) The first line of the grammar means "An *AExp* is an *AExp* plus an *AExp*, or an *AExp* minus *AExp*, or a term." Do the same thing for the other two lines by analogy.

(b) Why is "$*3$" not an *AExp*?

(c) Why is "$1+2+3$" an *AExp*?

(d) Draw a syntax tree for "$(1+2)*3$" and evaluate it.

(e) Draw a syntax tree for "$1 + 2 * x$." Compare this tree to the one from part (d).

2. Evaluating Sigma Notation.

In this section we will add a new kind of arithmetic expression to our grammar:

$$
\begin{array}{lcl}
AExp & a \stackrel{\text{def}}{=} & a_1 + a_2 \quad | \quad a_1 - a_2 \quad | \quad t \quad | \quad \sum_{x=n_1}^{n_2} f \\
term & t \stackrel{\text{def}}{=} & t_1 * t_2 \quad | \quad t_1/t_2 \quad | \quad f \\
factor & f \stackrel{\text{def}}{=} & (a) \quad | \quad r \quad | \quad n \quad | \quad x
\end{array}
$$

To evaluate, for each integer in the interval $[n_1, n_2]$, replace x in f for the integer, then add them together.

(a) Evaluate $\sum_{k=1}^{10}(k+1)$.

(b) Evaluate $\sum_{k=5}^{7}(k+1)$.

(c) Evaluate $\sum_{k=5}^{5}(k+1)$.

(d) Evaluate $\sum_{k=5}^{7} 1$.

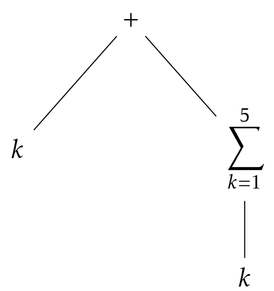
(e) Evaluate $\sum_{k=9}^{11}(k+1)^2$.

(f) Fill in the blanks for $\sum_{k=\_\_}^{\_\_} k^2$ so that this is the same sum as the one in part (e).

3. Sigma has complications: free and bound variables.

(a) Consider the function $f(k) = k + \sum_{k=1}^{5} k$. What do you think should be $f(1)$?

(b) The syntax tree for "$k + \sum_{k=1}^{5} k$" is



Each Σ (with variable k) *binds* all instances below the Σ, starting with the lowest one (like with evaluation). The remaining variables are *free. When substituting* 1 *for* k *in* $f(k)$, *we only substitute the free instances of* k.

(c) Either by drawing a syntax tree or just inspecting the expression, indicate the bindings (each k to which Σ) and free instances of the variable "k" in the expression

$$3 * k + \sum_{k=1}^{5} \left(k * \left(\sum_{k=1}^{5} (k * k) + 2 * \sum_{k=1}^{5} (k + 1) \right) \right).$$