Programming Languages and Computation

Week 2: Abstract Syntax Trees

We start with some revision of sets. If you are unfamiliar with any of the notation, please ask one of the TAs.

* 1. Write $\mathbb{N}$ for the set of natural numbers $0, 1, 2 \ldots$, and write Σ for the alphabet $\{0, 1\}$. List the elements of the following sets in any order.

(a) $\{1, 2, 3\} \cup \{2, 4, 6\}$

(b) $\{1, 2, 3\} \cap \{2, 4, 6\}$

(c) $\{1, 2, 3\} \times \{2, 4, 6\}$

(d) $\{1, 2, 3\} \setminus \{2, 4, 6\}$

(e) $\{2m \mid m \in \mathbb{N},\ 0 \leq m \leq 5\}$

(f) $\{uu \mid u \in \Sigma^*, |u| = 2\}$

(g) $\{u0v \mid u \in \Sigma^*,\ v \in \Sigma^*,\ |uv| = 2\}$

(h) $\{uvw \mid u \in \Sigma,\ v \in \Sigma,\ w \in \Sigma,\ w \text{ is the xor of } u \text{ and } v\}$

Suppose we have a countably infinite set of variable names, which we will just write as x, y, z, and so on. Recall the tree grammar for arithmetic expressions, where n can be any integer and x any variable name.

$$A \quad := \quad n \mid x \mid A+A \mid A-A \mid A*A$$

* 2. Draw the trees represented by the following inline expressions:

(a) $(45 + x) * 8$

(b) $45 + x * 8$

(c) $3 * 4 * 5$

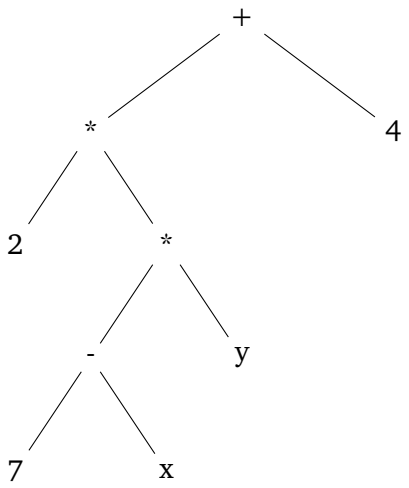
(d) $8 - 4 + 1$

(e) $180 * 2 - 4 + x * x$

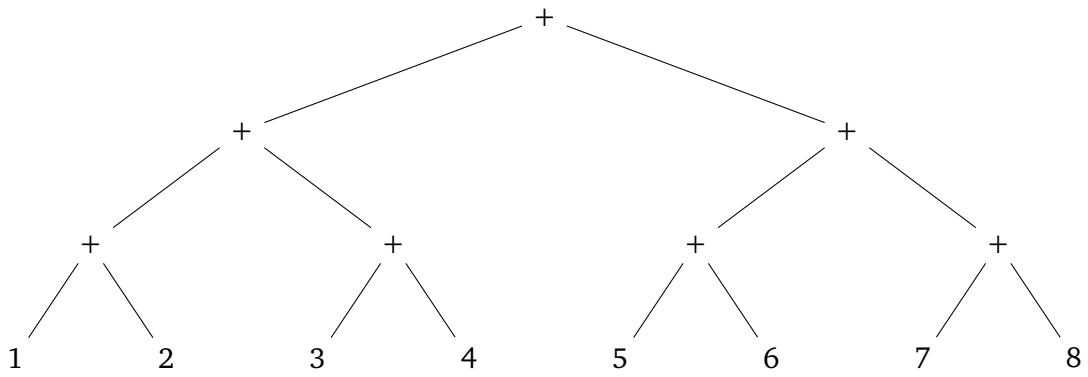
(f) $x * y + (7 - x) * 10$

* 3. Write the following abstract syntax trees as inline expressions:

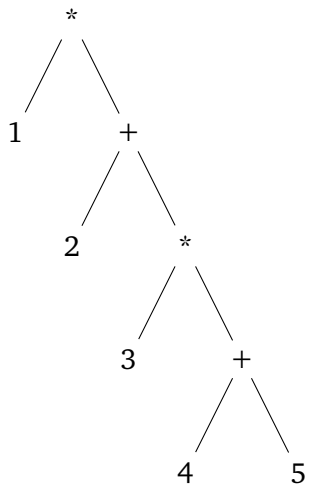
(a)



(b)



(c)



* 4. Show that each of the following trees is a valid arithmetic expression according to the abstract syntax described by the grammar above, by carrying out a derivation. Write all the syntax trees as inline expressions.

(a) $3 - 22$

(b) $1 + y$

(c) $1 + y * y$

(d) $(1 + y) * y$

(e) $66 - 2 * (x + y)$

The abstract syntax of Turtle programs can be defined by the non-terminal P in the following grammar, in which n is any natural number.

$$\begin{array}{lll} P & ::= & C \mid C\ ;\ P \\ C & ::= & \mathsf{up} \mid \mathsf{dn} \mid \mathsf{fd}\ n \mid \mathsf{lt}\ n \mid \mathsf{rt}\ n \end{array}$$

The tree constructors and their arities are:

n	0
up	0
dn	0
fd	1
lt	1
rt	1
;	2

We shall assume that the sequential composition operator ; associates to the right.

5. Give derivations for the following abstract syntax trees:

(a) dn

(b) dn ; up

(c) dn ; fd 20 ; up

* 6. Draw each of the trees given in the previous exercise explicitly.

** 7. Go to https://github.com/uob-coms20007/turtle and complete the tasks for Parts 0 and 1 of the Turtle compiler.