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

Solution

(a) $1, 2, 3, 4, 6$

(b) $2$

(c) $(1,2), (1,4), (1,3), (2,2), (2,4), (2,6), (3,2), (3,4), (3,6)$

(d) $1, 3$

(e) $0, 2, 4, 6, 8, 10$

(f) 0000, 0101, 1010, 1111

(g) 001, 100, 101, 000, 010, 011, 110

(h) 000, 011, 101, 110

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) $3*4*5$

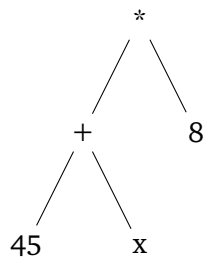
(d) $8-4+1$

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

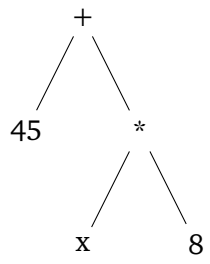
(f) $x*y+(7-x)*10$

Solution

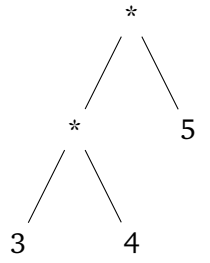
(a)



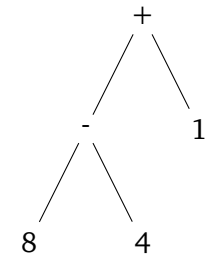

(b)



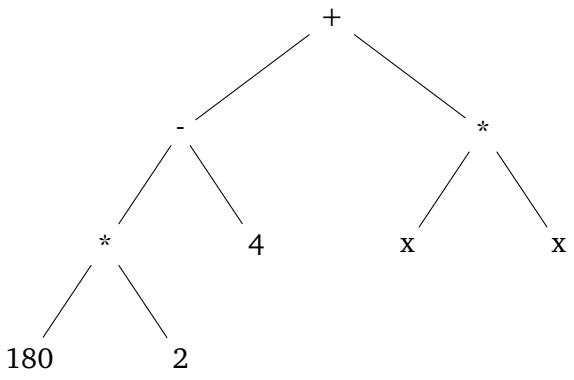

(c)



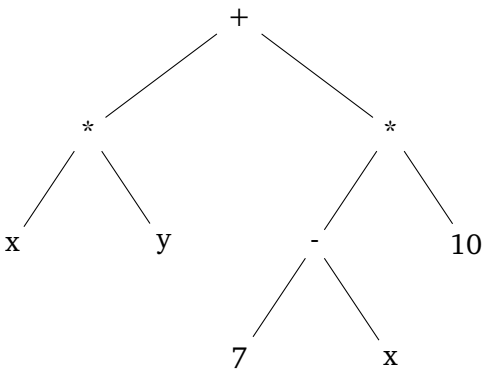
(d)




(e)

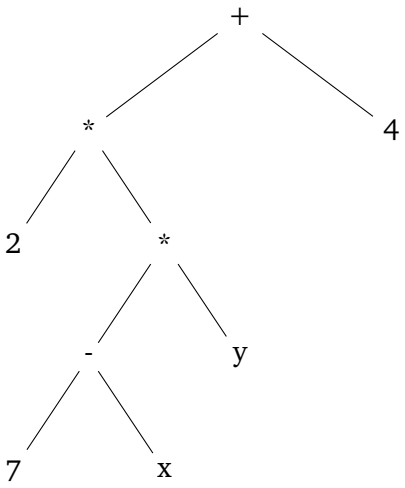



(f)

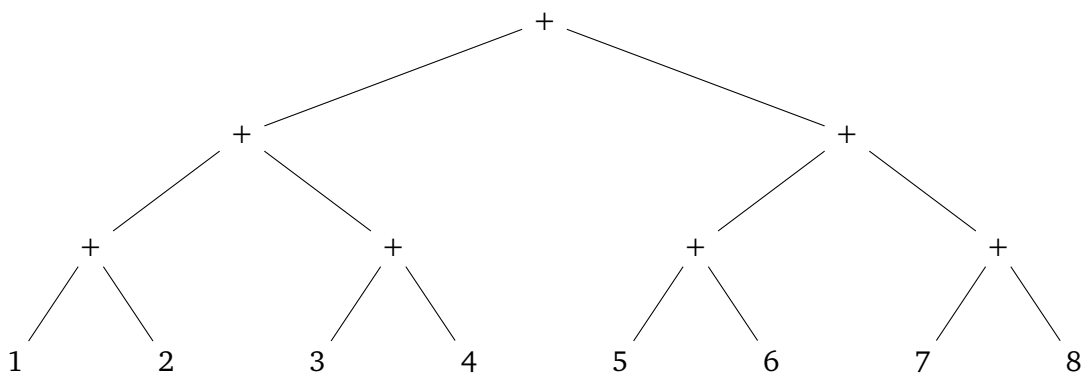



* 3. Write the following abstract syntax trees as inline expressions:

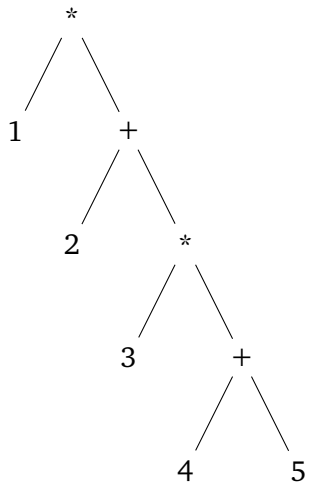
(a)




(b)




(c)




Solution

(a) $2 * ((7 - x) * y) + 4$

(b) $1 + 2 + (3 + 4) + ((5 + 6) + (7 + 8))$

(c) $1 * (2 + 3 * (4 + 5))$

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

Solution

There are different ways to carry out the derivation, because non-terminals may be chosen for replacement in a different order to the one given. However, the number of steps should be the same. You will not be penalised for using more parentheses than are strictly necessary according to the conventions.

(a)

$$
\begin{array}{rl}
 & A \\
\to & A-A \\
\to & 3-A \\
\to & 3-22
\end{array}
$$

(b)

$$
\begin{array}{rl}
 & A \\
\to & A+A \\
\to & 1+A \\
\to & 1+y
\end{array}
$$

(c)

$$
\begin{array}{rl}
 & A \\
\to & A+A \\
\to & 1+A \\
\to & 1+A*A \\
\to & 1+y*A \\
\to & 1+y*y
\end{array}
$$

(d)

$$
\begin{array}{rl}
 & A \\
\to & A*A \\
\to & (A+A)*A \\
\to & (1+A)*A \\
\to & (1+y)*A \\
\to & (1+y)*y
\end{array}
$$

(e)

$$
\begin{array}{rl}
 & A \\
\to & A-A \\
\to & 66-A \\
\to & 66-A*A \\
\to & 66-2*A \\
\to & 66-2*(A+A) \\
\to & 66-2*(x+A) \\
\to & 66-2*(x+y)
\end{array}
$$

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.

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

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

Solution

(a)

| | |
|---|---|
| | $P$ |
| → | $C$ |
| → | dn |

(b)

| | |
|---|---|
| | $P$ |
| → | $C$ ; $P$ |
| → | dn ; $P$ |
| → | dn ; $C$ |
| → | dn ; up |

(c)

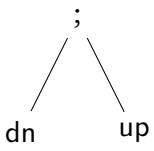
| | |
|---|---|
| | $P$ |
| → | $C$ ; $P$ |
| → | dn ; $P$ |
| → | dn ; $C$ ; $P$ |
| → | dn ; fd 20 ; $P$ |
| → | dn ; fd 20 ; $C$ |
| → | dn ; fd 20 ; up |
| → | dn ; fd 20 ; up |

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

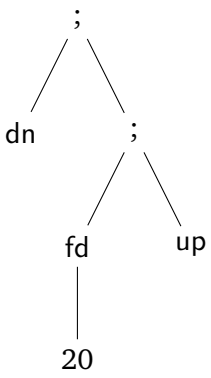
Solution

(a)


(b)




(c)




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