

CS 381 Grammar/AST – Practice

The following grammar defines the syntax of a sentence language.

$$\begin{array}{ll} n \in Noun & ::= \text{ dogs } | \text{ cats } | \text{ teeth } | \text{ fur } \\ v \in Verb & ::= \text{ have } | \text{ bite } \\ o \in Op & ::= \text{ and } | \text{ or } \\ s \in Sent & ::= \begin{array}{l} \text{neg } s \\ | \\ \text{op } o \ s \ s \\ | \\ n \ v \ n \\ | \\ v \ n \end{array} \end{array} \quad \begin{array}{l} \textit{negation} \\ \textit{binary operator} \\ \textit{noun verb noun} \\ \textit{verb noun} \end{array}$$

Binary operations are represented a bit differently than in other languages we have seen in class. To illustrate, here are two example expressions and their encoding in abstract syntax:

- dogs have fur and cats bite dogs op **and** (dogs have fur) (cats bite dogs)
- cats have teeth or cats have fur op **or** (cats have teeth) (cats have fur)

When you're drawing an abstract syntax tree, note that op is a node with three children – its operator (of type o) and two sub sentences (of type s).

1. Determine whether each of the following expressions can be generated from this grammar. In the blank provided write "Y" for *yes, it can be generated*, or "N" for *no, it cannot be generated*.

- (a) _____ and (cats bite dogs) (dogs have dogs)
- (b) _____ op and (cats have fur) (cats have teeth)
- (c) _____ op neg (op and (dogs bit cats) (dogs have teeth))
- (d) _____ op and (cats have cats) (dogs have dogs)
- (e) _____ op or cats (dogs have teeth)

2. For each of the following, draw the corresponding abstract syntax tree, or write “type error” if the expression cannot be generated from the grammar described on page 1.

(a) `neg (op or (cats have teeth) (dogs have fur))`

(b) `op and (neg (cats and dogs)) (have fur)`