**Syntax**

- What is a left/right associative function symbol?

Addition and subtraction are left associative

Exponentiation is right associative

- What are the infix, postfix and prefix notations of expressions?

Infix – operators in between operands

Postfix- operators after operands

Prefix- Operator before operands

- What is a BNF grammar?

A Grammar written in Backus Naur Form. Used to define the syntax of a language

- What is a metalanguage?

A language used to define a language, like SML, and BNF

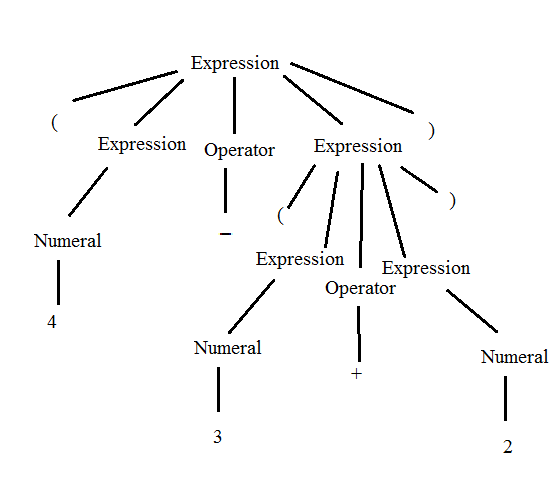
- What is a derivation (given a BNF grammar with a start symbol S)?

Example Derivation:

Sentence -> Subject Verb Object -> Subject Verb the Noun -> Subject Verb the cat -> Subject like the cat -> I like the cat

- What is a parse tree (given a BNF grammar with a start symbol S)?

---------------------------------------------------------------------------------------------------------------------



---------------------------------------------------------------------------------------------------------------------- -

Considering the grammar G defining the language L:

assignmentstatement -> variable = expression

expression -> variable | variable+variable

variable -> x | y | z

-Prove that x = y+z belongs to L using 1) a parse tree, and 2) a derivation.

AssignmentStatement -> Variable = Expression -> x = expression -> x = variable + variable -> x = y + variable -> x = y + z

- What is an ambiguous grammar?

A grammar where the same statement can be derived two different ways/have two different parse trees.

- How to remove ambiguity in grammars?

There is no set algorithm. Make a grammar that is equivalent but with unique parse trees

- What is the dangling else problem?

Where optional else clauses results in nested conditions being ambiguous (You’re not sure which if statement the else clause is attached to)

- What is a regular expression?

A sequence of characters/wildcards that matches a pattern

- Give a regular expression to represent expressions beginning with as, ending with bs and containing only as, bs and cs.

(as)((as)|(bs)|(cs))\*(bs)

- What is an EBNF grammar?

Extended Backus Naur Form. Backus Naur Form but with regular expressions

- What is/are difference/s between BNF and EBNF?

EBNF is a superset of BNF. EBNF allows wildcards

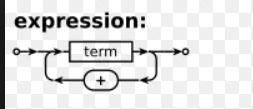
- Transform the BNF rule: Expression -> Term | Expression - Term | Expression + Term into a BNF rule?

- What are syntax diagrams?

Railroad diagrams/Syntax Diagrams are ways to express context-free grammar (syntax of a language).

- Give an example of syntax diagram?

----------------------------------------------------------



----------------------------------------------------------

- What are the phases of the compilation process? Describe each phase.

1: Lexical Syntax Analysis: It checks that each word in the program is a valid token

2: Concrete Syntax Analysis: It checks that the program for concrete syntax errors

3: Semantic Analysis: It checks the semantics of the program to make sure it makes sense (Not trying to add two booleans together,etc)

4: Intermediate Code Generation: turns the program into an intermediate phase between high level and machine language code. Like Bytecode

5: Optimization: Optimizes the code if the compiler is advanced enough to do so

6: Machine Code Generation: turns the program into machine code that can be executed by a computer.

- What are the differences between a compiler and an interpreter?

A compiler reads the entire program and converts it into computer-readable machine code. An interpreter reads a program line by line and executes it directly without any intermediate step

- What is Java bytecode?

Java code gets translated into bytecode which is then read by the JVM and converted to machine language. It is the intermediate phase of a java program

- What is the result of lexical analysis of a JAY program? (JAY program provided)

A lexical analysis returns the tokens of the program and its token types

A concrete analysis returns a parse tree

- What is the purpose of the lexical syntax? concrete syntax? abstract syntax?

Lexical syntax makes sure everything written in the program is defined in the grammar

Concrete Syntax checks the ordering of what’s written/ordering of tokens

Abstract Syntax looks at what the program is actually doing in terms of data types

- Transform an abstract syntax into a set of Java classes.

Create instance variables that correspond to the definitions in the abstract syntax. For example, Assignment -> Variable = Expression. In java it would be

Public class Assignment() {

Public Expression source;

Public Variable target; }

- Write a scanner.

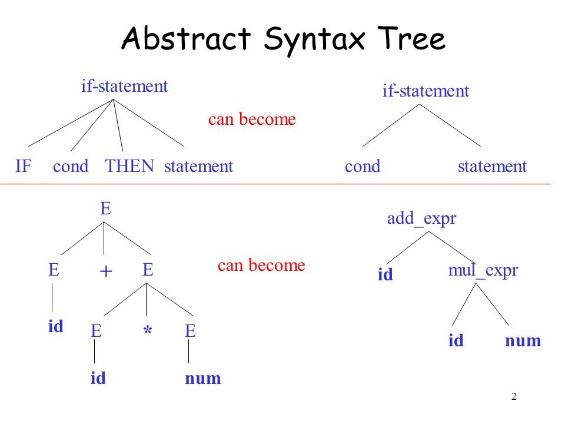
Check each token and assign it a type that is either one of the five tokens, or “other” meaning that it is a lexical error. Tokenstream.java

- Draw a concrete syntax tree with respect to a concrete syntax.

Start with Program which leads to the definition of a Program, and go keep going down the hierarchy until you reach the terminals.

- Draw an abstract syntax tree with respect to an abstract syntax.

An abstract syntax tree was not covered I believe. But it is something like this.



- Can 2 languages share the same concrete/abstract syntax?

Two languages cannot share the same concrete syntax. It would mean they are the same language.

- Describe the recursive descent parser algorithm.

The recursive descent parser algorithm shows how to write a parser for a concrete syntax by going through each of the cases that a concrete syntax definition can look like. (Defined by using terminals, non-terminals, a sequence of terminals/nonterminals, multiple possible definitions, etc)

- Implement a parser using the recursive descent parser algorithm.

Concretesyntax.java

- Give examples of programs with lexical, syntax and semantical errors.

Lexical Error in Jay: 4\_ ( \_ is not a legal symbol )

Syntax Error: int >; ( > is not an identifier, it is an operator )

Semantical Error: result = 3/0; (dividing by 0 doesn’t make sense )