Task 1: BNF ?

"BNF stands for Backus Naur Form it is a Formalization of a language that is to say that it is a map to all possible inputs in a language. In essence it is the constraints a given language has, this allows for a diagrammatic representation of a language, giving a visual representation of what is allowed at all levels in the language.

Task 2: BNF Description of L1s

<L1> ::= <P-List > | <Operation>

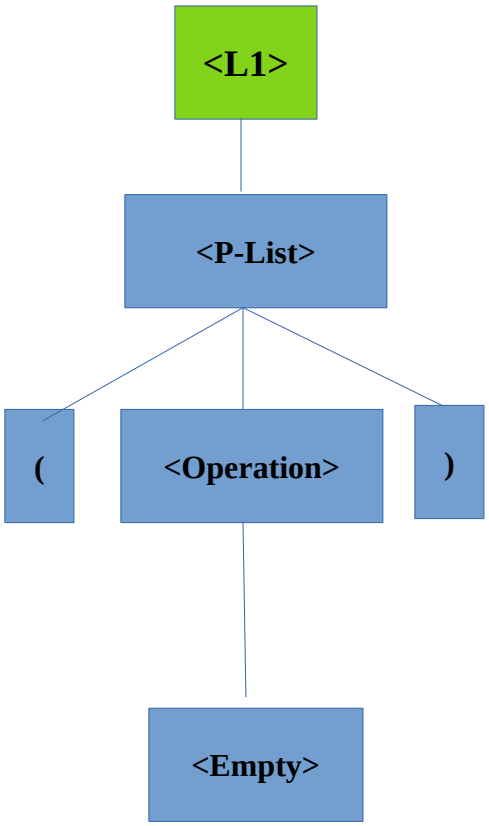
<P-List> :: = (<Operation>) | <empty>

<Operation>::= + <L1> | -<L1> | <empty>

Task 3 – Parse Tree for L1

---

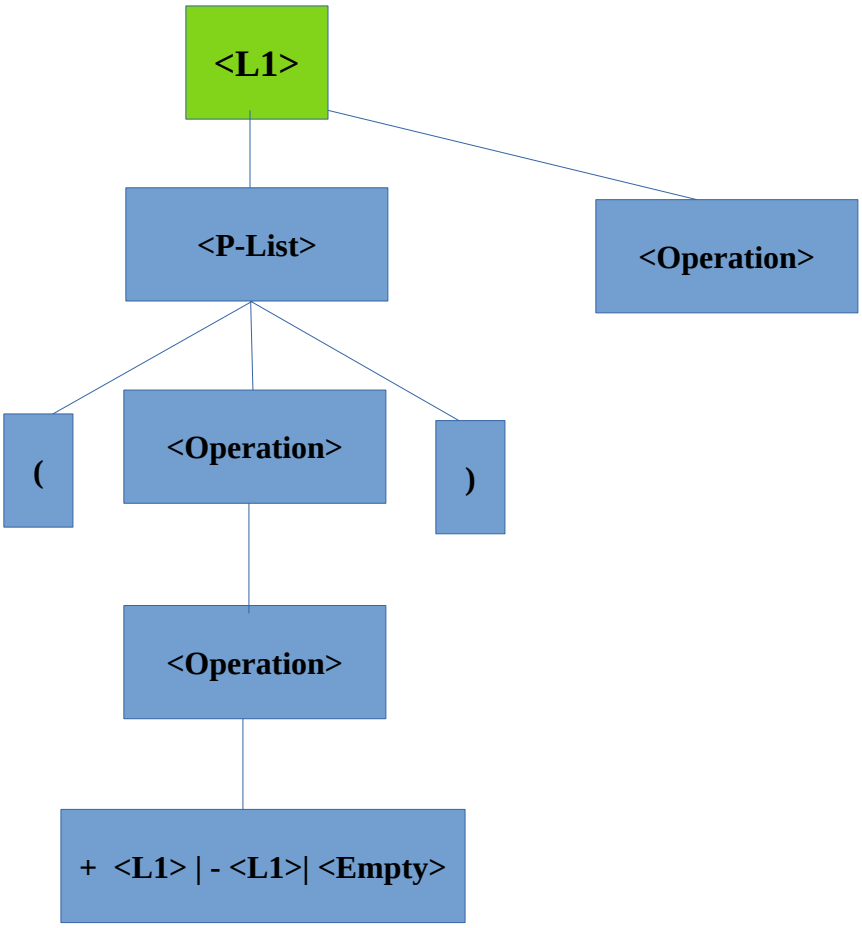
1. Parse Tree for **( )**



Task 3 – Parse Tree for L1 Cont.

---

2. Parse Tree for **(--)(++)**



Task 4– BNF Description of L2

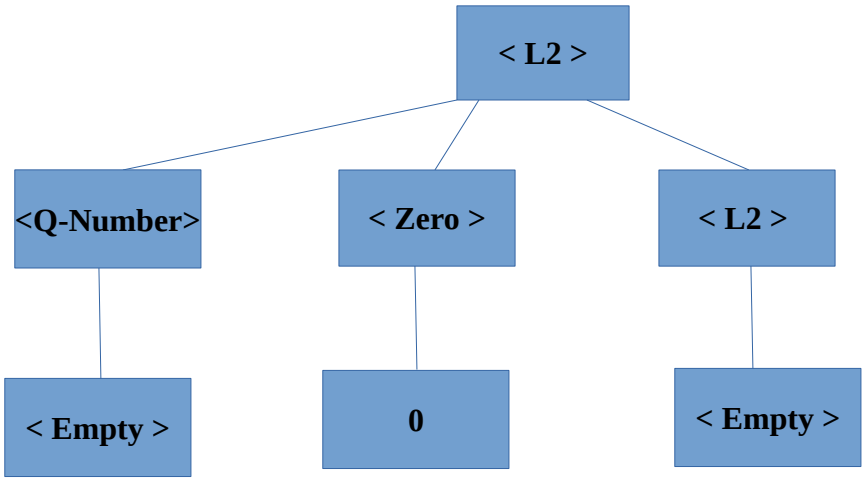
<L2> ::= < Q-Number > < Zero > < L2 > | < Empty >

<Q-Number> ::= 1 | 2 | 3 | < Empty >

<Zero> ::= 0 | < Q-Number > | < Empty >

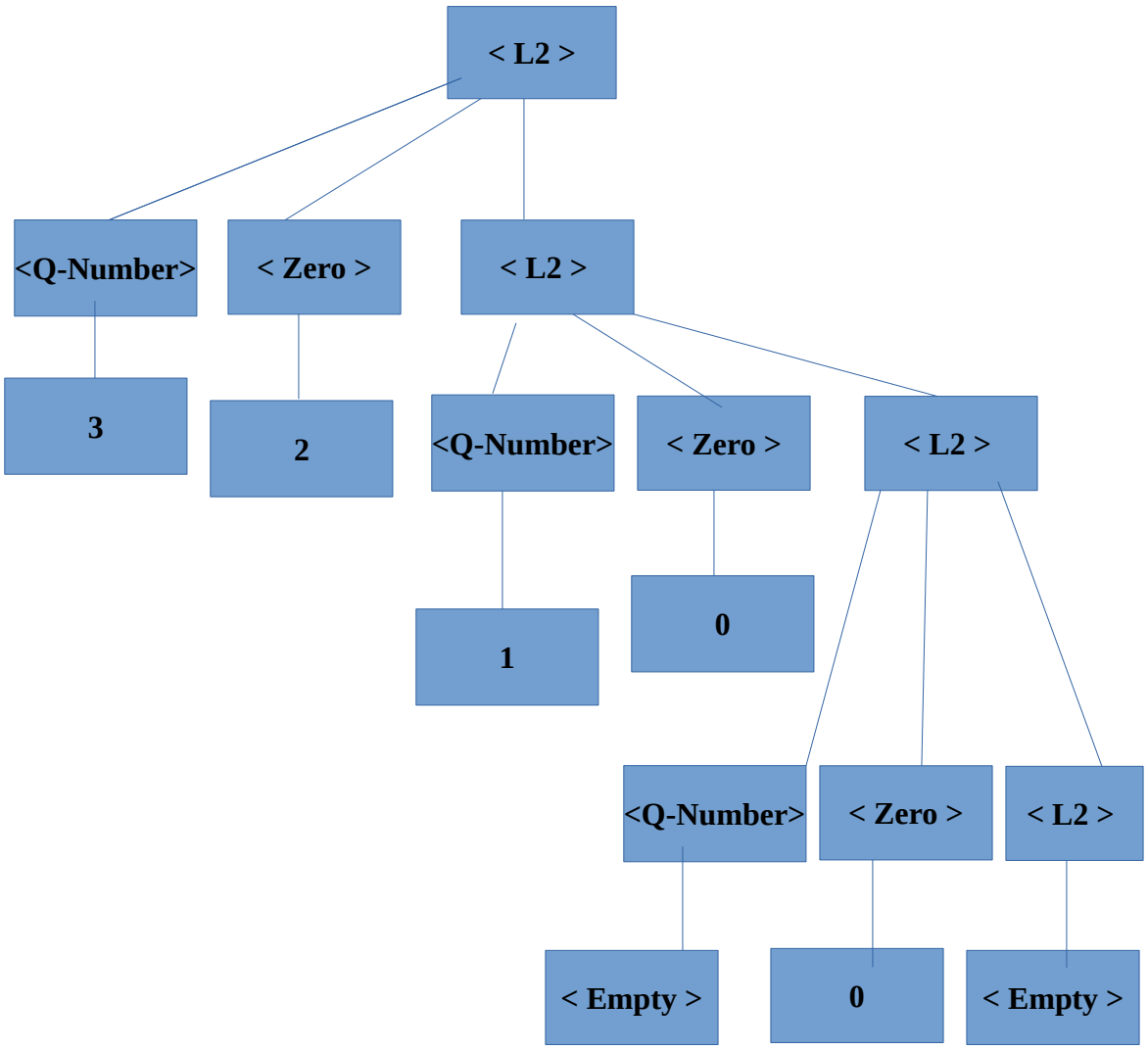
Task 5 - Parse Trees for L2

parse tree for 0



Task 5 - Parse Trees for L2

Parse for 32100



Task 6 - BNF Description of L3

< L3 > :: = < and > | < not > | < or > | < empty >

< and > :: = (and < TFR> < L3 >) | (and <L3 > <TFR>)

< TF > :: = #t | #f

<TFR> :: = < TF > | < TF > <TFR>

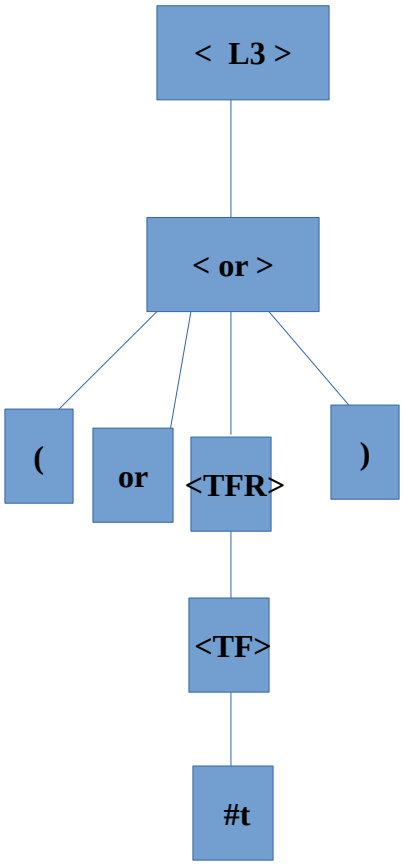
< or > :: = (or <TFR >) | (or < L3 >)

< not> :: = (not <TFR>)

Task 7 - Parse Trees for L3

---

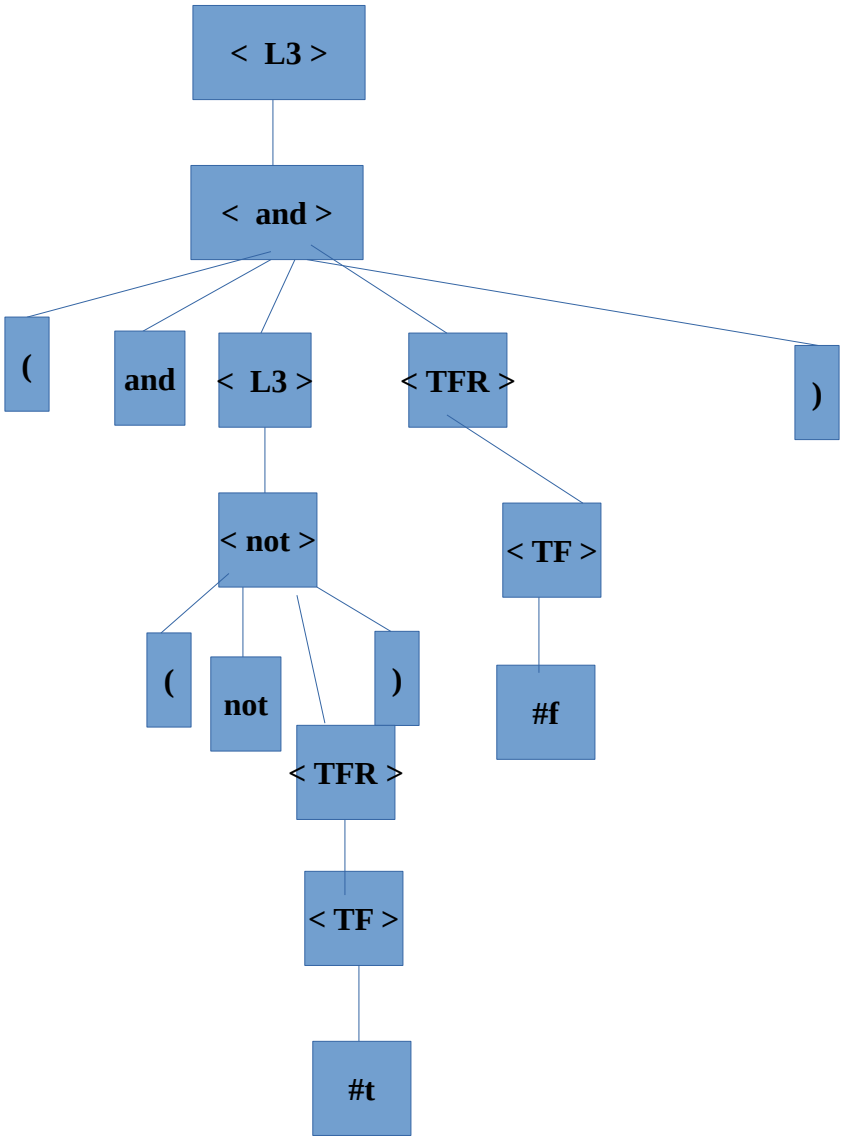
1. (or #t)



Task 7 - Parse Trees for L3

---

2. (and (not #t) #f)



Task 8 - BNF Description of L4

< L4 > :: = <oneD>|< tens>|<twoD> | <threeD>

< oneD > :: = zero | one | two | three | four | five | six | seven | eight | nine

< tens > :: = ten | eleven | twelve | thirteen | fourteen | fifteen | … | nineteen

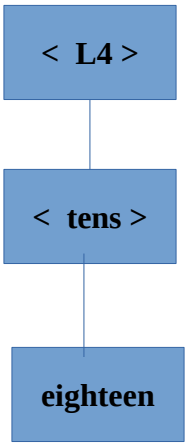
<twoD> :: = twenty <oneD> | thirty<oneD>| forty<oneD> | ...|ninety <oneD>

<threeD> :: = <oneD> hundred <oneD> | <oneD> hundred <twoD> | < oneD > hundred

Task 9 - Parse Trees for L4

---

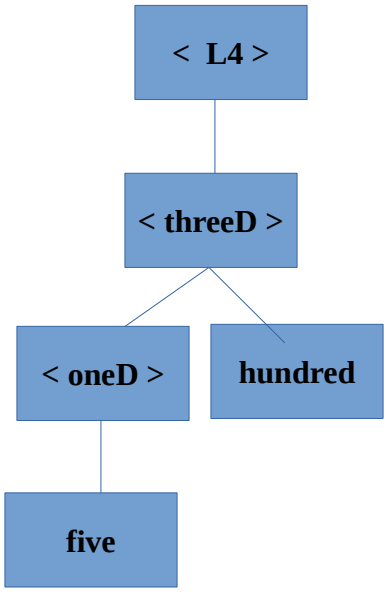
1. eighteen



Task 9 - Parse Trees for L4

---

2. five hundred



Task 9 - Parse Trees for L4

---

3. seven hundred fifty four