

Problem Set Specification

Task 1 - BNF?

Imagine that a computer science major asks you the question: “What is BNF?” Please write an answer in natural language, without examples, in a manner that you believe will serve to meaningfully inform the student about the nature and significance of BNF. Please do so in no more than 100 words.

Task 2 - BNF Description of L1

Consider language L1 to consist of the set of all strings of one or more parenthesized plus-strings or parenthesized minus-strings, where a parenthesized plus string consists of a left parenthesis followed by any number of plusses followed by a right parenthesis, and a parenthesized minus string consists of a left parenthesis followed by any number of minuses followed by a right parenthesis. Since English descriptions of languages like this invariably leave something to be desired, examples are generally provided in order to clarify the English specification. For L1, these examples should serve the purpose:

- [illegible]

Write a BNF grammar for this language.

Task 3 - Parse Trees for L1

Draw a parse tree for each of the following sentences in the L1 language.

1. ()
2. (--)(+++)

Task 4 - BNF Description of L2

Consider language L2 to be the set of all unsigned quaternary numbers (numbers composed of the symbols 0, 1, 2 and 3) with no leading zeros (that is, no zeros at the start of the number which are followed by a nonzero). Examples:

1. 0
2. 1001
3. 123000
4. 1020302010

Write a BNF grammar for this language.

Task 5 - Parse Trees for L2

Draw a parse tree for each of the following L2 sentences.

1. 0
2. 32100

Task 6- BNF Description of L3

Consider language L3 to be the set of Boolean valued expressions in Racket which are composed of the constants #t and #f and just the operators and, or and not. Here is a short Racket session that provides examples of L3 sentences:

```
Welcome to DrRacket, version 8.1 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> ( and #t ( not #f ) ( or ( not #t ) #f #t #f #t ) )
#t
> ( or ( and #t #t #t ) ( or #f #f #t ) )
#t
> #f
#f
> #t
#t
> ( and #f )
#f
> ( or #t )
#t
> ( and )
#t
> ( or )
#f
> ( not )
not: arity mismatch;
the expected number of arguments does not match the given number
expected: 1
given: 0
> ( not #t )
#f
> ( not #f #f )
not: arity mismatch;
the expected number of arguments does not match the given number
expected: 1
given: 2
>
```

Write a BNF grammar for this language.

Task 7 - Parse Trees for L3

Draw a parse tree for each of the following L3 sentences

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

Task 8 - BNF Description of L4

Consider language L4 to be the set English number names from 0 to 999 which contain no special symbols. Here is a suggestive look at this language:

- zero
- one
- two
- three
- four
- five
- six
- seven
- eight
- nine
- ten
- eleven
- twelve
- thirteen
- fourteen
- fifteen
- sixteen
- seventeen
- eighteen
- nineteen

- twenty
- twenty one
- twenty two
- ...
- thirty one
- thirty two
- ...
- ...
- ninety
- ninety one
- ninety two
- ...
- one hundred
- one hundred one
- one hundred two
- ...
- one hundred nineteen
- one hundred twenty
- one hundred twenty one
- ...
- nine hundred ninety seven
- nine hundred ninety eight
- nine hundred ninety nine

Write a BNF grammar for this language.

Task 9- Parse Trees for L4

Draw a parse tree for each of the following sentences in the L4 language.

1. eighteen
2. five hundred
3. seven hundred fifty four

Task 10 - BNF Description of L5

Please consider L5 to be a pseudonym for a little language called “Color Fun”. The full language can readily be inferred from the following demo by just looking at the lines beginning with the question mark prompt, and imagining reasonable generalizations in terms of RGB values. All of the sentences of L5 are represented, at least suggestively, in the demo.

Welcome to [DrRacket](#), version 8.1 [cs].
Language: racket, with debugging; memory limit: 128 MB.

```
> ( cf )  
? add ( 255 0 0 ) red  
? add ( 255 0 0 100 ) light-red  
? colors  
light-red red  
? show red
```



```
? show light-red
```



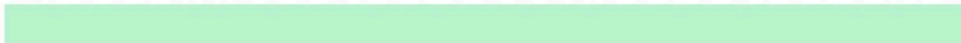
```
? describe red  
( 255 0 0 255 )  
? describe light-red  
( 255 0 0 100 )  
? add color c1  
? add color c2  
? add color c3  
? colors  
c3 c2 c1 light-red red  
? show c1
```



```
? show c2
```



```
? show c3
```



```
? describe c1  
( 60 66 228 11 )  
? describe c2  
( 132 41 143 164 )  
? describe c3  
( 91 230 141 116 )  
? add ( 100 150 200 ) c4  
? colors  
c4 c3 c2 c1 light-red red  
? show c4
```



```
? exit
```

Write a BNF grammar for this language.

Task 11 - Parse Trees for L5

Draw a parse tree for each of the following L5 sentences.

1. colors
2. show purple
3. add (100 220 170) c28

Task 12 - Document Compilation/Posting

Craft a nicely structured document that contains:

1. A nice title, indicating that this is your first problem set assignment, and that it focuses on BNF.
2. A nice learning abstract, which artfully says something about what you will be learning in this problem set.
3. A section that provides your BNF grammar for Task 2.
4. A section that provides the parse trees Task 3.
5. A section that provides your BNF grammar for Task 4.
6. A section that provides the parse trees Task 5.
7. A section that provides your BNF grammar for Task 6.
8. A section that provides the parse trees Task 7.
9. A section that provides your BNF grammar for Task 8.
10. A section that provides the parse trees Task 9.
11. A section that provides your BNF grammar for Task 10.
12. A section that provides the parse trees Task 11.