Csc344 BNF Assignment

What's It All About?

This assignment is all about BNF. You will be asked to compose some BNF grammars for given languages. You will be ask to draw some BNF parse trees. You will be asked to describe BNF in English, in a straightforward, compelling manner.

Big picture task - 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 focusses on BNF.
2. A nice learning abstract, which artfully says something about what you will be doing in this problem set, and what you will be learning in this problem set.
3. A section that provides your BNF grammar for Problem 1
4. A section that provides your BNF grammar for Problem 2
5. A section that provides your BNF grammar for Problem 3
6. A section that provides your BNF grammar for Problem 4
7. A section that provides your BNF grammar for Problem 5
8. A section that provides your BNF grammar for Problem 6

Post your document to you web work site.

Problem 1 - Shapes

The Language

Consider the language Shapes which consists of the set of all Lisp lists of length four, where:

- The first element is a list of length two, the first element of which is the token `size`, and the second element of which is either the token `large` or the token `medium` or the token `small`.
- The second element is a list of length two, the first element of which is the token `color`, and the second element of which is either the token `red` or the token `blue` or the token `yellow`.
- The third element is a list of length two, the first element of which is the token `pattern`, and the second element of which is either the token `striped` or the token `dotted` or the token `solid`.
- The fourth element is a list of length two, the first element of which is the token `shape`, and the second element of which is either the token `circle` or the token `square` or the token `triangle`.

For example, the following are sentences in this language:

1. `( ( size big ) ( color red ) ( pattern striped ) ( shape square ) )`
2. `( ( size medium ) ( color blue ) ( pattern dotted ) ( shape scircle ) )`
3. `( ( size small ) ( color yellow ) ( pattern solid ) ( shape triangle ) )`
4. `( ( size big ) ( color blue ) ( pattern solid ) ( shape circle ) )`

Tasks

1. Write a BNF grammar description of the Shapes language.
2. Draw a parse tree, consistent with the BNF grammar that you crafted, for the following sentence:
   `( ( size big ) ( color blue ) ( pattern solid ) ( shape circle ) )`
3. Draw a parse tree, consistent with the BNF grammar that you crafted, for the following sentence:
   `( ( size small ) ( color red ) ( pattern striped ) ( shape square ) )`

Problem 2 - SQN (Special Quaternary Numbers)

The Language

Consider the language SQN which consists of the set of all quaternary numbers with no leading zeros, and with no two adjacent occurrences of the same quaternary digit:

For example, the following are sentences in this language:

1. `0`
2. `123012301230`
3. `10121320212303132`

The following strings, for example, are **not** sentences in this language:

1. `1233`
2. `010`
3. `12322221`

Tasks

1. Write a BNF grammar description of the SQN language.
2. Draw a parse tree, consistent with the BNF grammar that you crafted, for the following sentence: `0`
3. Draw a parse tree, consistent with the BNF grammar that you crafted, for the following sentence: `132`
4. Explain, in precise terms, why you cannot draw a parse tree, consistent with the BNF grammar that you crafted, for the string: `1223`

Problem 3 - Fours

The Language

Consider the language Fours which consists of the set of all sequences of one or more occurrences of the following lists of digits, subject to the constraints that (1) any occurrances of `(1 1 1 1)` appear before any other lists, (2) any occurrences of (4) appear after any other lists, (3) any occurrences of lists containing a 2 occurs before any occurrences of lists containing a 3:

- `( 1 1 1 1 )`
- `( 1 1 2 )`
- `( 1 2 1 )`
- `( 2 1 1 )`
- `( 3 1 )`
- `( 1 3 )`
- `( 4 )`

For example, the following are sentences in this language:

1. `( 1 1 1 1 )`
2. `( 3 1 )`
3. `( 1 1 1 1 ) ( 2 1 1 ) ( 1 1 2 ) ( 1 2 1 ) ( 3 1 ) ( 1 3 ) ( 1 3 ) ( 3 1 ) ( 4 ) ( 4 )`
4. `( 4 )( 4 )( 4 )( 4 )`

Tasks

1. Write a BNF grammar description of the Fours language.
2. Draw a parse tree, consistent with the BNF grammar that you crafted, for the following sentence: `( 4 )`
3. Draw a parse tree, consistent with the BNF grammar that you crafted, for the following sentence:
   `( 1 1 1 1 )( 3 1 )( 1 3 )( 3 1 )`
4. Draw a parse tree, consistent with the BNF grammar that you crafted, for the following sentence:
   `( 1 1 1 1 )( 2 1 1 )( 3 1 )( 4 )`

Problem 4 - BXR

The Language

Consider language BXR 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 BXN 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
>
```

Tasks

1. Write a BNF grammar description of this language.
2. Draw a parse tree, consistent with the BNF grammar that you crafted, for the following sentence:
   `( or #t )`
3. Draw a parse tree, consistent with the BNF grammar that you crafted, for the following sentence:
   `( and ( not #t ) #f )`

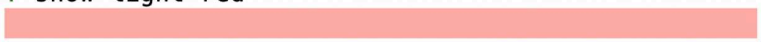
Problem 5 - CF (Color Fun)

The Language

Please consider CF 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 CF 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
Goodbye ...
>
```

Tasks

1. Write a BNF grammar description of this language.
2. Draw a parse tree, consistent with the BNF grammar that you crafted, for the following sentence: `colors`
3. Draw a parse tree, consistent with the BNF grammar that you crafted, for the following sentence: `show purple`
4. Draw a parse tree, consistent with the BNF grammar that you crafted, for the following sentence: `add ( 100 220 170 ) favorite-color`

Problem 6 - BNF?

Imagine that a freshman computer science major asks you the question: "What is BNF?" Please write an answer, **in natural language (English, please), 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.

Due Date

Friday, September 23, 2022