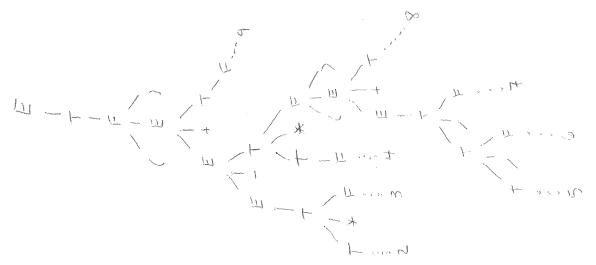
Practice - Context-free grammars (CF grammar)

1. Given the CF grammar for an expression, draw a parse tree for (2*3-4*(5/6/7+8)+9).

$$E \rightarrow E + T \mid E - T \mid T$$

$$T \rightarrow T * F \mid T / F \mid F$$

$$F \rightarrow (E) \mid << ident>> | << number>>$$



2. Write a BNF grammar for a Pascal case statement (which is similar to a C++switch). We are allowed nothing inside the case, i.e., an empty list. An example is the following:

```
case n1+n2 of
   5: write(blah, blah);
   6,7,8: read xxx, yyy;
   10,11: ...
end
```

Note that "case" and "of" and "end" are keywords. Note that the semicolon after the statement such as "write" or "read" is a separator, not a terminator. There cannot be a semicolon just before the "end".

Assume you are given to use without writing the non-terminals:

<expr> which generates arithmetic expressions such n1+n2,

<label> which generates one legal value such as 5 or 6 or 11 and so on, and

<stmt> which generates one statement such as a "write" (semicolon is not a part of <stmt>)

3. Write a CF grammar for expressions with balanced parentheses and square brackets. For example,

() [[]] ()()[] (()[()()][]) empty string Not legal: ([)] (()