C-lite Expression Grammar

```
exp       -> lvalue assignop exp | deref assignop exp | orterm

orterm    -> orterm || andterm | andterm

andterm   -> andterm && eqterm | eqterm

eqterm    -> eqterm equop relterm | relterm

relterm   -> relterm relop term | term

term      -> term addop factor | factor

factor    -> factor mulop unexp | unexp

unexp     -> lvalue autoop | & lvalue | * unexp | negop unexp | primary

deref     ->  * unexp

primary   -> ( exp ) | lvalue | constant | func

lvalue    -> var | ( lvalue )

var       -> id | id [ exp ]

func      -> id | id ( list )

list      -> exp | list , exp
```