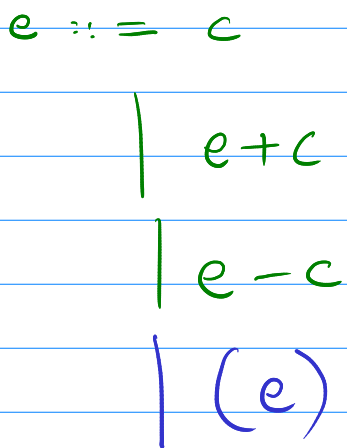
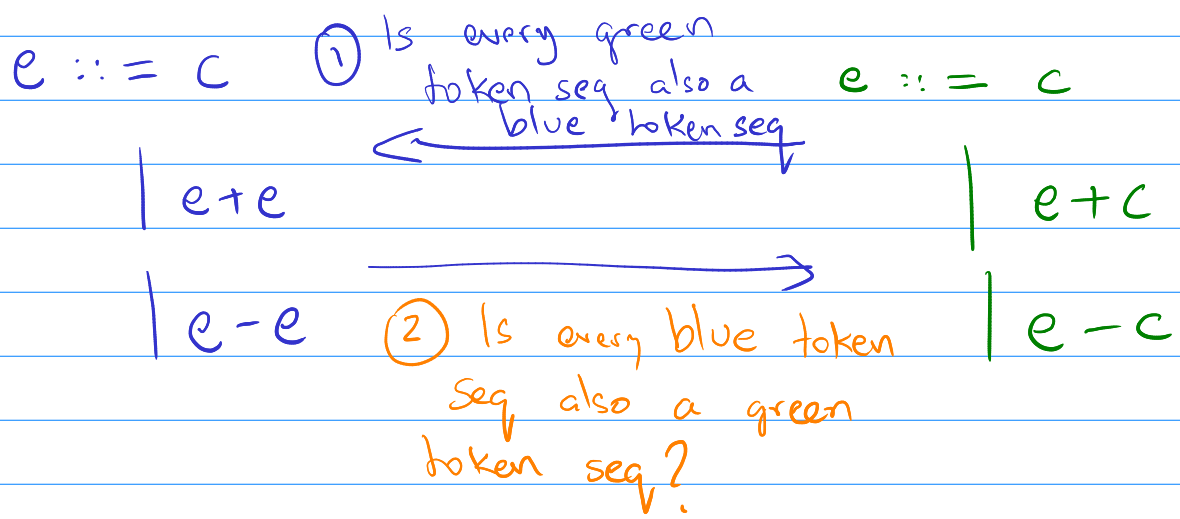


Ambiguous: There is some string which can be parsed in two different ways.



```

rule read =
  parse
  | [' '\t']+ $ read lexbuf }
  | '\n' { next_line lexbuf; read lexbuf }
  | ['0'-9']+ { INT_LIT (int_of_string (Lexing.lexeme lexbuf)) }
  | '+' { PLUS }
  | '-' { MINUS }
  | '(' { LPAREN }
  | ')' { RPAREN }
  | _ { raise (SyntaxError ("Unexpected character: " ^ Lexing.lexeme lexbuf)) }
  | eof { EOF }

```

<p style="color: red; margin: 0;">regular expressions.</p> <p style="margin: 0;">n</p> <p style="margin: 0;">$O(n)$ time</p> <p style="margin: 0;"> </p> <p style="margin: 0;">w $O(1)$ time</p>	<p style="color: blue; margin: 0;">context free grammars</p> <p style="margin: 0;">$O(n^3)$ time</p> <p style="margin: 0;">$O(n^2)$ memory</p>
---	--

Regular expressions

"Hi"

"HiHi"

"Hello"

"HiHiHi"

And I can repeat my greeting many times.

(Hi + Hello)*

I can greet someone with a hi

Or with a hello

I can greet them

(Hi + Hello)*

And then I can say good bye

(Bye)*

"Hi" ~ Hi

"Hello" ~ H;

Disjunction

(Hi + Hello) • (Bye + Ciao)

Concatenation

Kleene - *
0 or more
repetitions

(Hi)*

→ ""^ε

Hi

HiHi

HiHiHi