

# CS 131 Compilers: Discussion 7: Mid Term Preparation

杨易为 季杨彪 尤存翰

{yangyw,jiyb,youch}@shanghaitech.edu.cn

2021 年 4 月 12 日

## 1 Regular Expressions

1. Write a regex that matches binary strings divisible by 8.
2. Provide a regular grammar for the regex from part (a).

## 2 Finite State Automata

1. Write the corresponding NFA for the regular expression  $(0|1)^*(10|01)^+$ ;
2. Convert the NFA from part (a) into a DFA.

## 3 Grammar Rewriting and LL(k) Parsing

$$\begin{aligned} S &: E \mid \\ E &: E + E \\ &\mid E * E \\ &\mid ID \end{aligned}$$

1. Show that the grammar is ambiguous with two different leftmost derivations of the string  $a + b * c$ ;
2. Rewrite this grammar so that it preserves the standard order of operations, is LL(1), and is unambiguous. Draw the resulting tree for the string  $a + b * c$ ;
3. Write down the equivalent unambiguous grammar that enforces both left associativity and correct precedence. Why can't this be achieved with an LL(1) grammar?