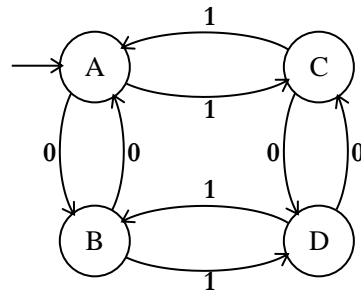


Principles of Compiler Construction

Quiz #5, June 10, 2021

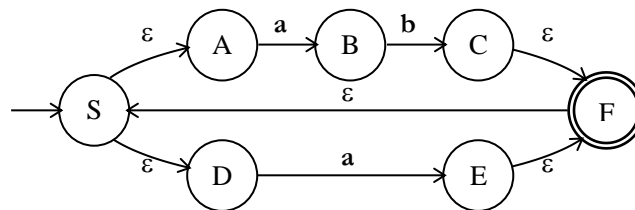
Part I (40 分)

设字母表 $\{0, 1\}$ 上的有限自动机 (Finite Automaton) 如下:



其中, A 是初始状态, 但目前暂时还没有任何终结状态。

- (1) (5 分) 该自动机是否一个确定的有限自动机 (DFA)? 为什么?
- (2) (5 分) 为让该自动机识别含有偶数个 **1** (含零个 **1**) 的所有串, 应将该自动机中的哪些状态改为终结状态?
- (3) (5 分) 为让该自动机识别长度为奇数的所有串, 应将该自动机中的哪些状态改为终结状态?
- (4) (10 分) 构造一个识别字母表 $\{0, 1\}$ 上至少含一个 **0** 且至少含有一个 **1** 的 DFA, 并解释该 DFA 的每一个状态的直观含义。注意不要忘记标注 DFA 的初始状态!
- (5) (15 分) 设有字母表 $\{a, b\}$ 上的 NFA 如下:



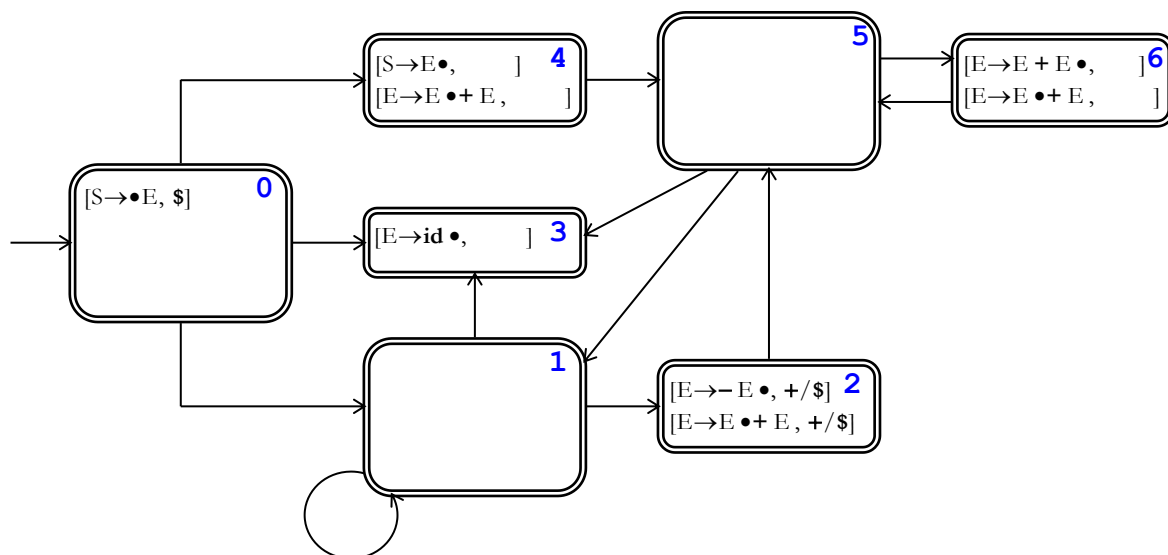
试将该 NFA 转换为等价的 DFA, 并在 DFA 状态中标明它对应的原 NFA 状态的子集。注意不要忘记标识 DFA 的初始状态!

Part II (60 分)

考虑以下文法：

$S \rightarrow E$
 $E \rightarrow E + E \mid -E \mid \text{id}$

以下是识别该文法所有活前缀的 DFA 的一个局部图：

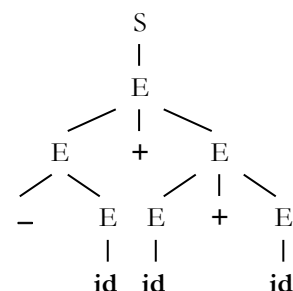


(1) (20 分) 补充完成上述 DFA，具体包括：计算状态 0 中已有有效项目的闭包并完成状态 0 的填写；填写状态 1 和状态 5 中的元素；填写状态 3、状态 4 和状态 6 中的向前看符号集；填写所有变迁上遗漏的符号。

(2) (5 分) 在该 DFA 含有归约项目的状态旁边标识 “reduce by P on x, y, \dots ”，表示在该状态见到 x, y, \dots 等向前看符号时用产生式 P 归约；对于接受状态则将 reduce... 改为 accept。

(3) (5 分) 对每一个含有冲突的状态，列出状态的编号、引起冲突的输入符号、以及冲突的类型（“移进—归约”冲突、“归约—归约”冲突）。

(4) (10 分) 显然，该文法是一个二义文法。假设我们想让句子 $-\text{id} + \text{id} + \text{id}$ 仅有如右图所示的这一棵分析树是合法的（以下将此称为性质 P ），请用自然语言描述：为保证性质 P ，相关算符的优先级和结合性质的规则如何？



(5) (10 分) 为保证性质 P ，根据上述 DFA 构造的 LR(1) 分析表中的冲突应如何解析？即在“移进—归约”冲突中选择移进还是归约、在“归约—归约”冲突中选择哪一个产生式归约？

(6) (10 分) 写出一个与原文法等价、但能够保证性质 P 的无二义文法。