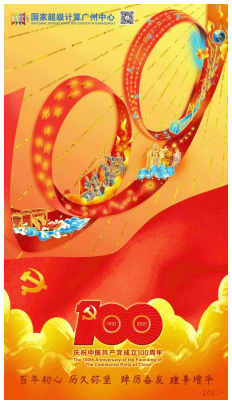
中山大學 SUN YAT-SEN UNIVERSITY

国家超级计算广州中心 NATIONAL SUPERCOMPUTER CENTER IN GUANGZHOU

Compilation Principle
编 译 原 理

Final Review



张献伟

xianweiz.github.io

DCS290, 7/1/2021

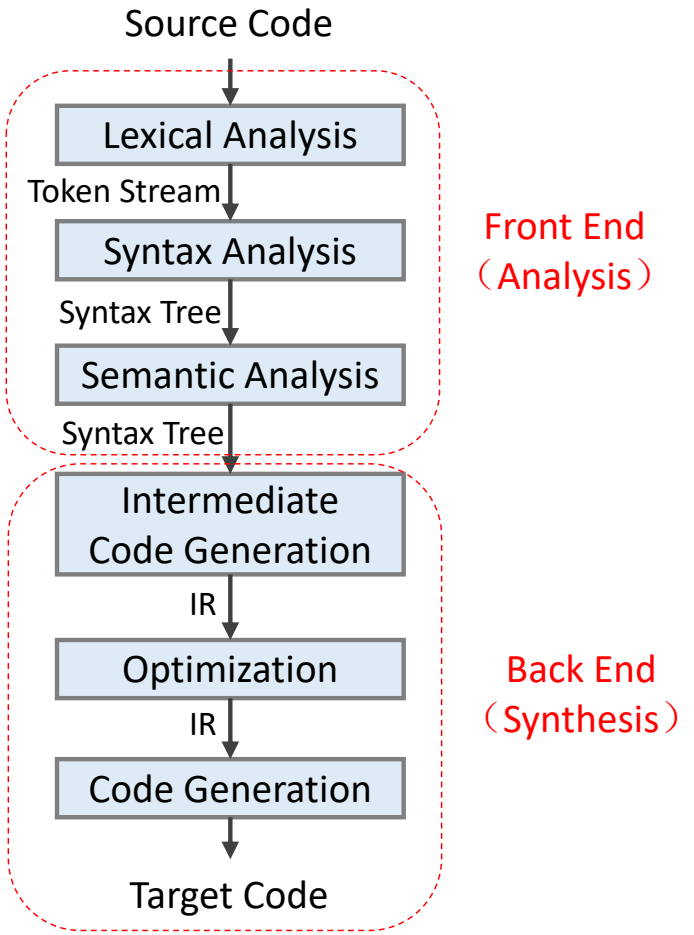
期末考试

- 时间：7/8(周四), **14:30 – 16:30**
- 地点：**A105**

- 试题语言：中文（关键术语标注英文）
- 分值：100分（占总成绩60%）
  - 前端：70%
  - 后端：30%
- 题型：
  - 判断题（8x1’）
  - 简答题（2x12’）
  - 应用题（3x16’）
  - 综合应用题（1x20’）

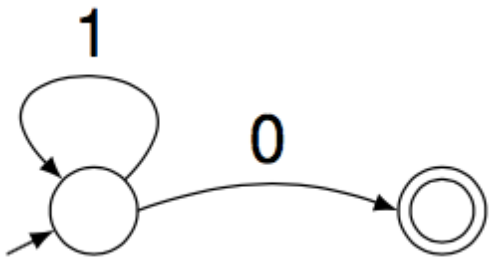
Compilation Phases[编译阶段]

- **Lexical**: source code → tokens
  - RE, NFA, DFA, …
- **Syntax**: tokens → AST or parse tree
  - CFG, LL(1), LALR(1), …
- **Semantic**: AST → AST +symbol table
  - SDD, SDT, typing, scoping, …
- **Int. Code Generation**: AST → TAC
  - IR, offset, CodeGen, …
- **Optimization**: TAC → (optimized) TAC
  - BB, CFG, DAG, …
- **Code generation**: TAC → Instructions
  - Instruction, register, stack, …



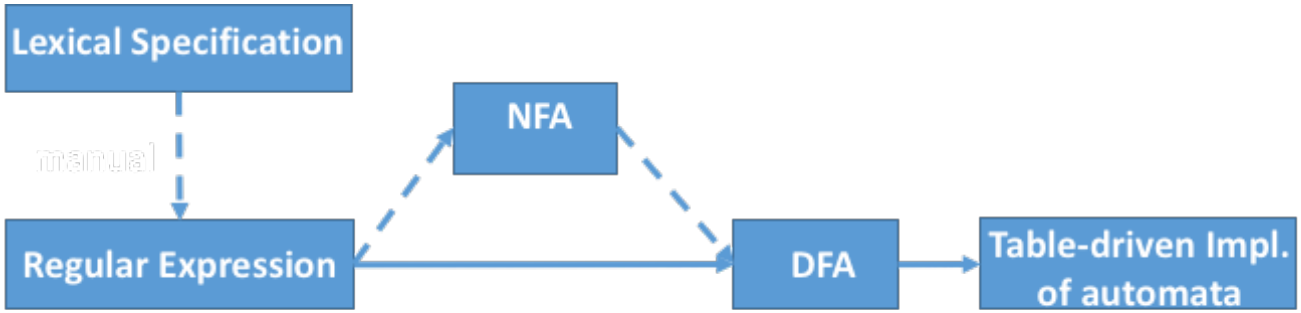
Lexical Analysis[词法分析]

- Characters --> tokens
  - 二元组：<class, lexeme>
- How to specify tokens?
  - Regular expression
    - Atomic, compound
- How to recognize tokens?
  - Transition diagram[转换图]
  - NFA, DFA, table



1*0

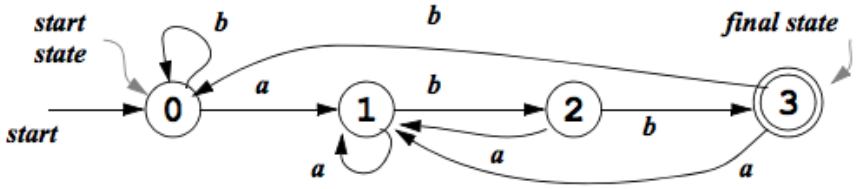
Any number of ‘1’s followed by a single ‘0’



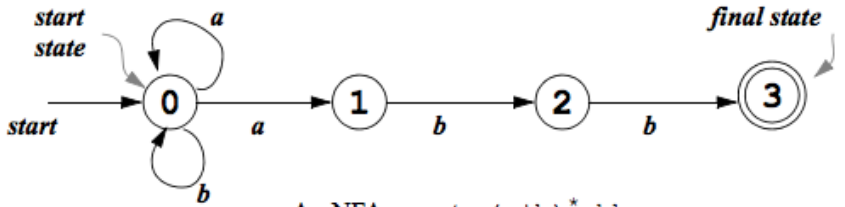
Lexical Analysis (cont.)

- Regular expression
  - 自然语言描述 <-> 正则表达式
  - 正则表达式 <-> NFA/DFA
  - 局限性： RE vs. CFG
    - $L = \{a^n b^n \mid n \geq 1\}$

- NFA, DFA
  - 状态和边的含义（ε-move）
    - 初始状态、终结状态
  - 形式上的区别
  - 意义上等价
    - NFA -> DFA: ε-闭包
    - 状态最小化



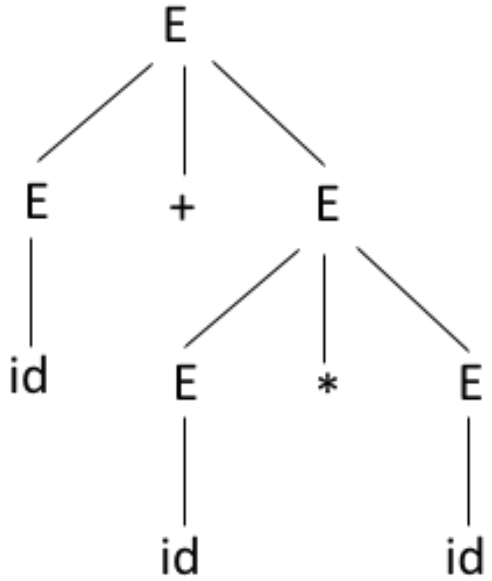
A DFA accepts (a|b)*abb



An NFA accepts (a|b)*abb

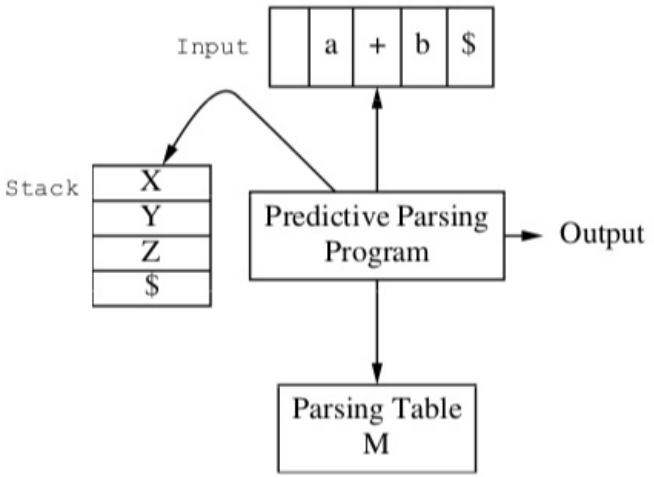
Syntax Analysis[语法分析]

- Tokens --> parse tree
- Context-free grammar
  - 四元组：T, N, s, σ
    - 但通常只写σ
  - Production rule: LHS --> RHS
  - 所定义语言的自然描述
- Derivation
  - Leftmost, rightmost
  - Parse tree: 推导的图形化表示
  - Sentential form[句型]、Sentence[句子]
  - Ambiguity二义性及消除



Syntax Analysis (cont.)

- Parser
  - Top-down: leftmost derivation
  - Bottom-up: reverse order of the rightmost derivation
- Top-down
  - Recursive descent, Predictive/LL(k)
  - Left recursion: rewriting
  - Common prefix: left factoring
- LL(1)
  - Build the parse table
    - FIRST, FOLLOW
  - Use the parse table: expand or match
    - 给定输入串的分析过程
  - Determine if G is LL(1)



Syntax Analysis (cont.)

- Bottom-up
  - Shift-reduce
  - Handle[句柄]、Viable Prefix[活前缀]、Phrase[短语]、Simple Phrase[直接短语]
    - 活前缀不能越过句柄：分析栈存放的都是活前缀，在等句柄出现；一旦出现就规约这个句柄
    - 句柄是一个直接短语

令 G 是一个文法，S 是文法的开始符号，假定 αβδ 是文法 G 的一个句型，如果有

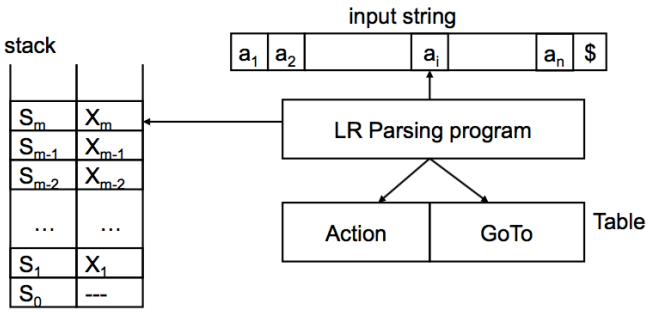
$$S \overset{*}{\Rightarrow} \alpha A \delta \text{ 且 } A \overset{+}{\Rightarrow} \beta$$

则称 β 是句型 αβδ 相对于非终结符 A 的短语。特别是，如果有

$$A \Rightarrow \beta$$

则称 β 是句型 αβδ 相对于规则 A→β 的直接短语，一个句型的最左直接短语称为该句型的句柄。

- LR
  - More powerful than LL
  - Parse table:
    - Action table: shift/reduce/accept/error
    - Goto table
  - LR分析器的工作过程实际上就是逐步产生规范句型的活前缀
    - 构造识别所有活前缀的DFA == 构造基于项目集的DFA
  - 给定parse table，对输入串进行分析



Syntax Analysis (cont.)

- LR(0): build parse table, construct a FA
  - Item/configuration: initial, reduce, accept
  - State/configuration set: closure()
  - Augmented grammar
  - DFA -> parse table
  - Conflicts: shift-reduce, reduce-reduce
- Other LRs
  - SLR
    - Improve LR(0): FOLLOW
  - LR(1)
    - LR(1) item: LR(0) item + lookahead symbols
    - Configuration set: closure()
  - LALR(1)

Semantic Analysis[语义分析]

- Semantic attributes
  - Synthesized, inherited
- Semantic rules or actions
- SDD vs SDT
  - Syntax directed definitions: attributes + semantic rules
    - S-attributed, L-attributed
  - Syntax directed translation scheme: attributes + semantic actions
    - An executable specification of the SDD
- Annotated parse tree
  - With actions

Code Generation, Optimization[后端]

- Intermediate representation
  - Three-address code
  - CodeGen: variable, array, control, …
- Runtime/Target code
  - Stack, AR, calling conventions
  - Operations via registers ($fp, $sp, $ra, …)
- Code optimizations
  - Concepts: basic block, flow graph, DAG
  - Optimization: metrics, techniques
    - Dead code elimination, common subexpression elimination
    - Strength reduction, constant folding, …

参考资料

- 考试样题练习
  - final_practice: https://xianweiz.github.io/teach/dcs290/hw_projs/final_practice.pdf
- 作业及期中练习
  - hw1 solution: https://xianweiz.github.io/teach/dcs290/hw_projs/hw1_sol.pdf
  - hw2 solution: https://xianweiz.github.io/teach/dcs290/hw_projs/hw2_sol.pdf
  - mid solution: https://xianweiz.github.io/teach/dcs290/hw_projs/mid_sol.pdf
- 随堂review questions和quiz
  - 详见课件：https://xianweiz.github.io/teach/dcs290/s2021.html
- 其他
  - https://github.com/ysyisyourbrother/SYSU_Notebook/tree/master/编译原理%20常会友
  - https://github.com/sysuexam/SYSU-Exam/tree/master/编译原理