**Introduction**

**Master**: the phases of a compiler, compilers vs. interpreters

**Understand**: what is a compiler

You Should be familiar with each stage of the compiler, and describe the whole process in graph or text, and can tell the difference between a compiler and an interpreter.

**Lexical**

**Master**: Write regular expression(RE), the transition from REs to NFAs, then to DFAs, as well as from FAs to REs, DFA minimization and the construction of scanner

**Understand**: Concept of regular expression, NFA, DFA

Be familiar with the expression of RE, Know the difference of RE and CFG. Given a RE, know how to construct its NFA and DFA, or vice versa, and how to minimize it.

**C.F.G.**

**Master**: Context-free grammar(CFG), Derivation/reduction, leftmost derivation and rightmost derivation, sentential form and sentence, Parse tree, Abstract syntax tree, grammars with left recursion and/or left factor, ambiguous grammar

Be familiar with the grammar of CFG, know how to make a left/right most derivation. Given a set of grammar and an expression, can draw the parse tree and abstract syntax tree*.* Can recognize ambiguous grammars. Able to eliminate the left factor and left recursion in grammars

**Top-Down Parsing**

**Master**: what is the predictive parsing? LL(1) grammar, First set and Follow set, Recursive-descent parsing, LL(1) parsing(non-recursive parsing), parsing table

**Understand**: error recovery in predictive parsing.

Given the CFG, know how to compute its first/ follow set, judge if it is a LL(1) grammar，construct the parsing table, and parsing the input based on the parsing table step by step.

**Bottom-Up Paring**:

**Master**: Right sentential form, Handle, Viable prefix, items and states, Shift/reduce(LR) parsing, LR(0) parsing, SLR(1) parsing

**Understand**: LR(1) parsing and LALR(1) parsing

It must be easy of you to write out the right sentential form of a given CFG and expression, and figure out the variable prefix and handles. Know the difference of LR(0) and SLR(1). Knowing when will the conflicts happen. What's more, given the CFG, be able to write out its LR(0) items, construction the LR(0) DFA, construct the LR(0)/SLR(1) parsing table, and know how to parse a input according to parsing table step by step.

**Semantic**:

**Master**: Attributes and Attribute grammar(SDD), Synthesized and Inherited attributes. S-Attributed grammar and L-Attributed grammar, Dependency graphs and evaluation of attributes

**Understand**: The purpose of semantic analysis, what is type checking

Know the task of semantic analysis, and the implementation method of it. Master how to construct a dependency graph, and compute the values of the attributes according to the attribute grammars, and build the annotated parsing tree. Understand why we talk about S-Attributed grammar and L-Attributed grammar.

**Intermediate Code Generation**:

**Master**: Three-address code, SDT: intermediate code generation for expression, statement, and control flow structure (Branch and loop structures)

**Understand:** IR and several types of IR we have learned

Given SDD, can translate the source code into TAC by SDT, more importantly, Boolean expressions in control flows.

考试题型 1）选择题 10题共20分；2）解答题 共6题每题5分至25分不等

题目顺序与难度无关。先做会做的，切勿在一个地方卡死，不要简单问题复杂化。