

Department of Computer Science and Information Engineering
 PhD Qualify Exam – Programming Languages and Compilers, Fall 2010

1. (20%) For each of the following language types, give a concise description of the main characteristics and name TWO existing languages of that type.

Language Type	Characteristics	Examples
Procedural languages		
Declarative languages		
Imperative languages		
Object-oriented languages		
Logic programming languages		
Functional languages		
Markup languages		
Scripting languages		
Rule-based languages		
Macro languages		

2. (14%) For the following questions, use only the core regular expression notations $A+B$, AB , A^* , A^+ , ϵ , and “abc”.

2(a) Write a regular expression over the alphabet $\Sigma = \{a, b, c\}$ for the language of strings that

- start and end with a
- contain an odd number of b
- and any number of c in between

2(b) Draw a DFA over the alphabet $\Sigma = \{a, b\}$ for the language of non-empty strings that contain an even number of a or an odd number of b .

3. (16%) For each of the following statements on programming language concepts, circle TRUE or FALSE based on whether the statement is ALWAYS true or not.

TRUE or FALSE	For each regular expression r and the corresponding language $L(r)$, there exists a DFA that accepts $L(r)$.
TRUE or FALSE	For each regular expression r and the corresponding language $L(r)$, there exists an NFA that accepts $L(r)$.
TRUE or FALSE	Given a context-free grammar g and the corresponding language $L(g)$, there exists a DFA that accepts $L(g)$.
TRUE or FALSE	A well-formed XML document must be a valid document.
TRUE or FALSE	A valid XML document must be a well-formed document.
TRUE or FALSE	A language can either be a compiled or interpreted language, but not both.
TRUE or FALSE	A program written in an interpreted language is always slower than its compiled language counterpart because every time it is executed, it must be reinterpreted again.
TRUE or FALSE	A compiler-compiler is the compiler to compile the source program of a compiler.

4. (5%) How can the one-pass compiler work? How can the multiple-pass compiler work?

5. (10%) Please construct the follow set, first set, predict set and parse table of the following grammar.

```

<program>      -> begin <statement list> end
<statement list> -><statement> {<statement>}
<statement>    ->ID:=<expression>;
<statement>    ->read ( <id list>);
<statement>    ->write (<expr list>);
<id list>      ->ID {, ID}
<exp list>     -><expression> {, <expression>}
<expression>  -><factor> { <add op> <factor>}
<factor>      -><primary> {<mult op> <primary>}
<primary>     ->ID
<primary>     ->INTLITERAL
<primary>     ->(<expression>)
<add op>      ->PLUSOP
<add op>      ->MINUSOP
<mult op>     ->MULTOP
<mult op>     ->DIVIDOP

```

<system goal> -><program> SCANEOF

6. (5%) Please write the parse tree of the following codes according to the modified grammar in question 5.

begin read A; B:=A+3; write B; end SCANEOF

7. (20%) Please construct the LR(1) machine, goto table, and action table of the following grammar, and trace the parsing steps performed by the LR(1) parser on the input ((ID@,(ID@,ID@)),ID@)\$ and show the parsing steps.

$S \rightarrow A\$$

$A \rightarrow (B,A)$

$A \rightarrow C@$

$B \rightarrow B,A$

$B \rightarrow A$

$C \rightarrow ID$

8. (10%) Please construct the LR(0) machine, goto table, and action table of the following grammar, and trace the parsing steps performed by the LR(0) parser on the input {{ID=ID+ID};ID=ID+ID} \$ and show the parsing steps.

$P \rightarrow B\$$

$B \rightarrow \{S\}$

$S \rightarrow S;E \mid E$

$E \rightarrow B \mid ID=E \mid ID+ID$