Compiler Technology of Programming Languages
Final Exam

Instructor: Farn Wang
Class hours: 09:10-12:00 Tuesday Room: BL 114

Student name: Student ID:

1. Please explain what run-time compilation is. (3points/3)

2. What advantages does the run-time compilation technology bring out ? (3points/6)

3. Please describe a use case where run-time compilation brings out advantages. (4points/10)

4. Please write down a regular expression for variable names in C language. (5points/15)

5. Please draw the finite-state automata for recognizing the regular expression for the last question (i.e, question 3). (5pts/20)

6. Please write down the CFG for integer array declaration of arbitrary dimensions in C. (5points/25)

7. Please draw the parse tree according to your CFG in question 6 for the following array declaration. (5points/30)

int a[1][3], b[3][4][5][6];

8. Please write down a CFG that is in LL(2) but not in LL(1). You must explain the correctness of your answer with an example sentence in your CFG. (5points/35)

9. Please write down a CFG that is in LR(0) but not in LL(k) for any k. You must explain the correctness of your answer with an example sentence in your CFG. (10points/45)

10. Please write down a CFG that is in LR(1) but not in LR(0). You must explain the correctness of your answer with an example sentence in your CFG. (5points/50)

11. Please draw the AST (abstract syntax tree) for the following program segment. (8points/58)

```
a = 3; c = a; s = 0;
while (c > 0) { s = s + c*c; c--; }
printf("%5d: %10d\n", a, s);
```

12. Please explain the advantages of using AST as an intermediate representation for code generation instead of using parse tree. (2points/60)

13. Please explain the reachability analysis and termination analysis in Java interpreter. Please write down an example Java code segment to make your explanation. (5points/65)

14. Please explain the activation record (AR) layout with stack pointer (SP) and frame pointer (FP) in code generation phase of compilation. (5points/70)

15. We have machine code for an assignment statement A=B*B+C*C in the following.

A = B*B + C*C (assuming only 3 available regs)

	Register Association		
	W9	W10	W11
ldr w9, B	B (NS)		
mul w10,w9,w9		Temp(S)	
ldr w11, C			C(NS)
mul w9,w11,w11	Temp(S)		
add w11,w9,w10	freed	freed	A(S)

The register association steps are also shown in the table above. The machine instructions are explained in the following.

ldr r, B:	*load memory content at address B to register r.*
str r, B:	*store the content of register r to memory cell at address B.*
add ri, rj, rk:	*add the contents of register rj and rk and save the result in register ri.*
mul ri, rj, rk:	*multiply the contents of register rj and rk and save the result in register ri.*

Here **ldr** and **str** are memory instructions while **add** and **mul** are register instructions. Assume that we still only have 3 registers W9, W10, and W11 to associate with data items in code generation. Please write down a machine code segment for the following assignment statement: A=B*C + B*(C+A) so that minimum number of memory instructions are generated. If necessary, you may create temporary variables in memory. To justify the correctness of your answer, you must also draw the register association table as in the above example. (10points/80)

16. In the 42nd ACM PLDI 2021 paper: *Snapshot-free, Transparent, and Robust Memory Reclamation for Lock-free Data Structures*, the authors proposed a new reference counting scheme for dynamic memory reclamation. Reference counting schemes are usually considered high-overhead technologies and impractical for garbage collection. What techniques did the authors propose to reduce the overhead? What is the assumption for the correct application of their techniques? (5points/85)

17. In the 42nd ACM PLDI (2021) paper: *Developer and User-Transparent Compiler Optimization for Interactive Applications*, the authors proposed a capture/replay scheme for online optimization of machine code for mobile apps. Please briefly explain the core idea of their technology. (5points/90)

18. In the 50th ACM ICPP (2022) paper: *EasyView: Enabling and Scheduling Tensor Views in Deep Learning Compilers*, the authors discussed the interaction between tensor views and operation fusion to optimize code for machine learning and evaluation. What is a tensor view? What is operation fusion? How can they together be used to enhance the performance of machine learning modules? (5points/95)

19. In the 49th ACM POPL (2022), there is a paper: *Truly Stateless, Optimal Dynamic Partial Order Reduction*. Please explain the technology of partial order reduction and its application for speeding up program analysis. Please explain the technical difference between dynamic and static partial order reductions. Please explain the concepts of TSO and PSO. (5points/100)