IN4303 — Compiler Construction

Exam

January 18, 2011

---

1. Answer every question on its own page.
2. Put your name, your student ID and the number of the question on top of each page.
3. If you need more than one page to answer a question, number your pages and state the overall number of pages for this question on the first page.
4. Take care of your time. This exam has 12 questions, for a total of 150 points. Try to answer a question worth 10 points in 10 minutes.
5. Keep your answers short and precise. Don't waste your time on essay writing.
6. Hand in the answers together with this form (including the questions).
7. When you finished the exam, please do the evaluation survey.

Good luck!

Name: _____ Student ID: _____

Question	Points	Score
Properties of language	10	
Formal grammars	10	
Syntax Definition Formalism *SDF*	10	
Term Rewriting	10	
Static analysis	5	
Java Bytecode	10	
Polymorphism	10	
Stack frames	15	
Register allocation	35	
Garbage collection	5	
Lexical analysis	15	
LR parsing	15	
Total	150	

Question 1: Properties of language (10 points)

(a) Explain five properties of language using the example of English. (5)

(b) Explain why a software language like MiniJava meets these properties as well. (5)

Question 2: Formal grammars (10 points)

Let G_1 be a formal grammar with nonterminal symbols S and E, terminal symbols $\mathbf{x}$, $+$, $*$, (and), start symbol S, and the following production rules:

$$\begin{aligned} S &\rightarrow E \\ E &\rightarrow \mathbf{x} \\ E &\rightarrow E + E \\ E &\rightarrow E * E \\ E &\rightarrow (E) \end{aligned}$$

(a) Is G_1 context-free? Why (not)? (1)

(b) Describe the language defined by G_1 in English. (2)

(c) Give a right-most derivation for the sentence $\mathbf{x} + (\mathbf{x} * \mathbf{x})$ according to G_1. (3)

(d) Give two different parse trees for the sentence $\mathbf{x} + \mathbf{x} * \mathbf{x}$ according to G_1. (4)

Question 3: Syntax Definition Formalism *SDF* (10 points)

Compared to plain context-free grammars, *SDF* provides additional description means for the syntax of formal languages. Explain two of these additional description means and why they are needed in compiler construction.

Question 4: Term Rewriting (10 points)

Stratego provides a strategy `map` with the following implementation:

```
map(s): []     -> []
map(s): [x|xs] -> [<s> x | <map(s)> xs]
```

(a) What is the result of applying `map(inc)` to the term `[1,2,3]`? (2)

(b) Explain the semantics of `map` in English. Illustrate your explanation by an example. (4)

(c) Explain the concept of recursion based on the definition of `map`. (4)

Question 5: Static analysis (5 points)

(a) Why do we need static analysis in a compiler? (1)

(b) Name resolution and type analysis are two typical static analysis. Explain both using the example of MiniJava. (4)

Question 6: Java Bytecode (10 points)

Execute the following bytecode instructions, starting with an empty stack.

```
    bipush 21
    ldc 5
    iconst_4
    dup
    imul
    iadd
    isub
    ifeq l1
    iconst_1
l1: nop
```

Show the stack after each instruction. If an instruction performs a jump, name the next instruction.

Question 7: Polymorphism (10 points)

The following Java program includes several examples for polymorphism:

```
public class A {}

public class B extends A {}

public class C {
    public int m() { return 1 + 2; }
}

public class D{
    public String m() { return "1" + "2"; }

    public A m(A a1, A a2) { return a1; }
}

public class E extends D {
    public A m(B b1, B b2) { return b2; }
}
```

Identify five examples of polymorphism in the program. Which kinds of polymorphism do they represent? Explain the differences.

Question 8: Stack frames (15 points)

A Java class `A` defines a method `int m(int p1, int p2)`.

(a) Which steps (on bytecode level) are needed to call `m(1,2)` on an object `a` of type `A`? Which steps are needed to return from a call? (5)

(b) Explain how the Java Virtual Machine handles the call and the return. (5)

(c) Explain *single dispatch* as used by the Java Virtual Machine. Compare this approach with two other kinds of dispatch. (5)

Question 9: Register allocation (35 points)

Consider the following intermediate code.

```
    c1 := r3          e := e - 1
M[42] := c1           if e > 0 goto l1
    a := r1           r1 := d
    b := r2           c2 := M[42]
    d := 0            r3 := c2
    e := a            return
l1: d := d + b
```

(a) Construct the control graph. (2)

(b) Calculate successor nodes, defined variables, and used variables for each node in the control graph. (3)

(c) Assume `r1` and `r3` to be live-out on the return instruction. Calculate live-ins and live-outs for each node in the control graph. (12)

(d) Construct the interference graph. Include special edges for move instructions. (5)

(e) Assume `r1`, `r2`, and `r3` to be precoulored, i.e. they interfere with each other. Colour the interference graph with three colours. When coalescing two nodes, say whether you are using the Briggs or George criterion. (10)

(f) Rewrite the original program according to the colouring by using `r1`, `r2`, and `r3` as registers. Remove redundant move instructions. (3)

Question 10: Garbage collection (5 points)

(a) What is the basic assumption of generational collection? (2)

(b) How is it performed? (3)

Question 11: Lexical analysis (15 points)

Consider the following regular expression

$$r_1 \quad = \quad (\mathbf{A}|\ldots|\mathbf{Z})^*(\mathbf{0}|\ldots|\mathbf{9})^*$$

(a) Describe the language defined by r_1 in English. (2)

(b) Turn r_1 into an equivalent finite automaton. (4)

(c) Remove ε-moves from the finite automaton. (4)

(d) Is the resulting automaton deterministic? Why (not)? (1)

(e) Use the automaton to generate a word with at least five letters. Enumerate the states passed during the generation. (2)

(f) Use the automaton to recognise the word **IN4303**. Enumerate the states passed during the recognition. (2)

Question 12: LR parsing (15 points)

Let G_2 be a formal grammar with nonterminal symbols S, T and E, terminal symbols $\mathbf{x}$, $+$ and \$, start symbol S, and the following production rules:

$$\begin{aligned} S &\rightarrow E\ \$ \\ E &\rightarrow T + E \\ E &\rightarrow T \\ T &\rightarrow \mathbf{x} \end{aligned}$$

(a) Explain the role of the terminal symbol \$. (1)

(b) Construct a LR(0) parse table for the grammar. (8)

(c) What kind of conflict does the resulting parse table contain? (1)

(d) Explain two strategies to resolve this conflict. (2)

(e) Assume the conflict is resolved in favour of a shift. Use the parse table to recognise the sentence $\mathbf{x} + \mathbf{x}$. Show the stack and the remaining input after each step. (3)