IN4303 — Compiler Construction

Exam

January 31, 2017

---

1. You can make each question in WebLab or on paper.
2. When you answer a question in WebLab, make sure you save and submit the answer. Most questions come with several subquestions. Make sure you save each of your answers separately. Answers will be lost, if you only submit at the end of the page.
3. When you answer a question on paper, answer each question on its own page (each sheet has four pages).
4. Put your name, your student ID and the number of the question on top of each page.
5. If you need more than one sheet to answer a question, number your pages and state the overall number of pages for this question on the first page.
6. Take care of your time. This exam has 10 questions, for a total of 150 points. Try to answer a question worth 10 points in 10 minutes.
7. Keep your answers short and precise. Do not waste your time on essay writing. If you cannot answer a particular question, admit it and move on.
8. Hand in the answers together with this form (including the questions).

Good luck!

Name: ______________________ Student ID: ______________________

Question	Points	Score
Language specification	5	
Formal Grammar	10	
Lexical analysis	20	
LR parsing	20	
Term rewriting	10	
Name resolution	20	
Java Virtual Machine	10	
Dataflow Analysis	20	
Register allocation	15	
Garbage collection	20	
Total	150	

Question 1: Language specification (5 points)

(a) What is a *language workbench*? (1)

(b) Name three different aspects of a programming language, which need to be specified or implemented. Explain, what each aspect is about. (2)

(c) How can each of the different aspects of a programming language be formally specified? (2)

Question 2: Formal Grammar (10 points)

Let G be a formal grammar with nonterminal symbols E and B, terminal symbols **id**, $+$, $-$, $*$, and $/$, and the following production rules (annotated with constructor symbols in the style of SDF3):

$$\begin{array}{rcl} E.I & \rightarrow & \mathbf{id} \\ E.B & \rightarrow & E\ B\ E \\ B.P & \rightarrow & + \\ B.S & \rightarrow & - \\ B.M & \rightarrow & * \\ B.D & \rightarrow & / \end{array}$$

(a) Demonstrate that this grammar is ambiguous by giving two different abstract syntax trees for the sentence `id + id * id` and the left-most derivations that give rise to these trees. (3)

(b) Rewrite the grammar in SDF3 notation and use declarative priority and associativity rules to disambiguate it following standard rules. Define a translation from the ASTs of this grammar to the AST (3)

(c) Instead of using separate disambiguation rules, transform the grammar to an unambiguous context-free grammar. (4)

Question 3: Lexical analysis (20 points)

Consider the following regular expression

$$r_1 \quad = \quad (\mathbf{0}|\ldots|\mathbf{9})^+(.(\mathbf{0}|\ldots|\mathbf{9})^+)^?(\mathbf{E}(\mathbf{0}|\ldots|\mathbf{9})^+)^?$$

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

(b) Transform the regular expression r_1 to the equivalent expression r_2 that does not use the ? and + operators (4)

(c) Systematically construct an equivalent finite automaton for the r_2 expression (4)

(d) What does equivalence in this case mean? (1)

(e) Transform the automaton to an equivalent automaton without ε-moves. (4)

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

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

(h) Use the automaton to recognize the word **3.14E12**. Enumerate the states passed during the recognition. (2)

Question 4: LR parsing (20 points)

Let G be a formal grammar with nonterminal symbols S and E, terminal symbols **id**, $+$, (,) and \$, start symbol S, and the following production rules:

$$\begin{array}{rcl} S & \rightarrow & E\ \$ \\ E & \rightarrow & \mathbf{id} \\ E & \rightarrow & \mathbf{id}\ (\ E\ ) \\ E & \rightarrow & E\ +\ \mathbf{id} \end{array}$$

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

(b) Draw a diagram with the LR(0) states and transitions for grammar G (8)

(c) Build the LR(0) parse table for grammar G (5)

(d) Is this an SLR grammar? Why (not)? (2)

(e) Use the parse table to recognize the sentence `id(id(id) + id)$`. Show the stack and the remaining input after each step. (4)

Question 5: Term rewriting (10 points)

Consider the following algebraic signature representing the abstract syntax of an expression language in *Stratego* :

```
signature
  constructors
    Var : String -> E
    Int : Int -> E
    Add : E * E -> E
    Mul : E * E -> E
    And : E * E -> E
    Or  : E * E -> E
    If  : E * E * E -> E
```

In this language integers are used to represent Boolean values, with `Int(0)` representing false and all other integers representing true. The Boolean operators `And` and `Or` are short-circuit operations.

(a) Define in Stratego a desugaring transformation to eliminate the `And` and `Or` operators. (2)

(b) Give the term resulting from desugaring the term `And(Var("x"), Or(Int(1), Var("y")))` (2)

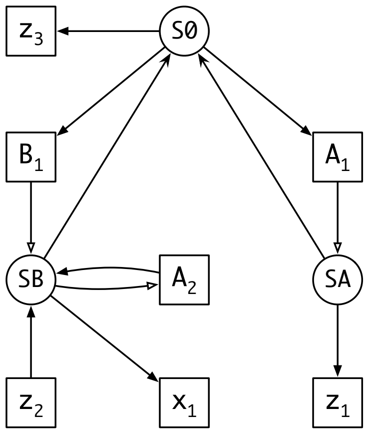
(c) Define in Stratego a strategy that, given a transformation that maps variables to expressions, applies this transformation to all variables in an expression; use only basic operators. (2)

(d) Define in Stratego a constant folding transformation for desugared expressions. (4)

Question 6: Name resolution (20 points)

(a) Explain how the scope graph approach to name resolution works. (5)

(b) Given the following scope graph, list all reachability paths for reference z_2 and indicate the path selected according to visibility rules. Motivate your answer. (5)



(c) Given the following Java program, draw its scope graph.Note that the subscripts of identifiers indicate the position of the identifier in the program. Two occurrences x_i and x_j denote different occurrences of the same name. (5)

```
class A_1 {
  static int i_2;
  static int f_3(int i_4) {
    return i_5;
  }
  static int g_6(int i_7) {
    return f_8(B_9.i_10);
  }
}
class B_11 {
  static int i_12;
  static int h_13(int j_14) {
    return A_15.g_16(i_17);
  }
}
```

(d) For each reference in the program list its resolution path and the declaration to which it resolves. (5)

Question 7: Java Virtual Machine (10 points)

Execute the bytecode instructions of `Main/main()V`, starting with an empty stack. The initial value of local variable 0 is `4242 4303`, pointing to an object of class `Foo`. Show stacks and local variables after each instruction. If you have to invoke a method, execute the bytecode of this method as well. Make clear when stack frames are created and destroyed, and which data is passed between frames.

`Main/main()V`

```
    aload_0
    bipush 4
    iconst_2
    isub
    invokevirtual Foo/f(I)V
```

`Foo/f(I)V`

```
    iload_1
    goto l2
l1: iload_1
    ldc -2
    iadd
    dup
    istore_1
l2: ifne l1
    return
```

Question 8: Dataflow Analysis (20 points)

Consider the following intermediate code:

```
1       c := r3
2       a := r1
3       b := r2
4       d := 0
5       e := a
6  l1:  d := d + b
7       e := e - 1
8       if e > 0 goto l1
9       r1 := d
10      r3 := c
11      return
```

(a) Construct the control graph for the shown intermediate code on paper. You are allowed to omit the actual code and to use line numbers instead. (2)

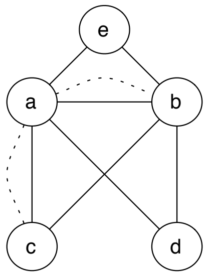
(b) Provide a table with successor nodes, defined variables, and used variables for each node in the control graph. (3)

(c) Calculate live-ins and live-outs for each node in the control graph in a table on paper. Show each round of calculation. (15)

Question 9: Register allocation (15 points)

Liveness analysis of the intermediate code on the left yields the interference graph on the right, which should be coloured with two colours.

```
s: if a = 0 goto r
   e := b / a
   d := e * a
   c := b - d
   b := a
   a := c
   goto s
r: return b
```



(a) Can both move-related edges be removed safely? Why (not)? (1)

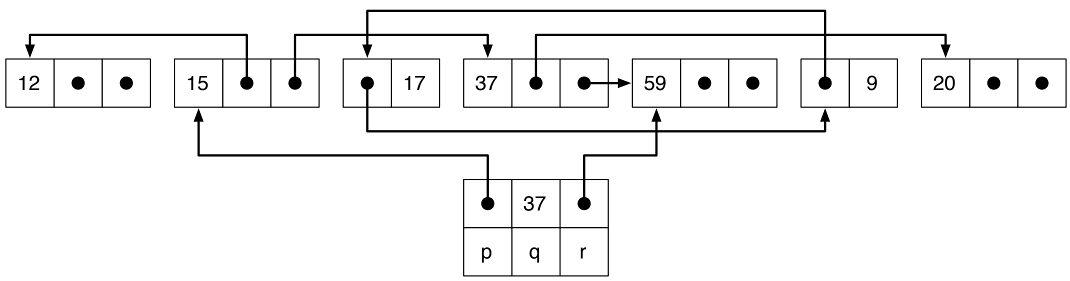
(b) Assume the move-related edges were removed. Why does the next step need to be a spill? (2)

(c) Spill node `a` and continue the graph colouring until you can decide if this spill is an actual one. (7)

(d) Is node `a` an actual spill? (1)

(e) Perform the spill on the intermediate code. (4)

Question 10: Garbage collection (20 points)



(a) What kind of data is stored on the heap? (2)

(b) Perform a copy collection on the given heap data structure. Show the data structure after each copying step (after pointer adjustments). (15)

(c) Explain the key benefit of copy collection over mark-and-sweep garbage collection. (1)

(d) Explain the effect of copy collection on the locality of data. (2)