Mälardalen University
School of innovation, design and engineering

TEN1 DVA339 2020

Compiler theory

Thursday, January 16, 2020, 14:10 - 18:30
Contact: Daniel Hedin, 021-107052 (15:00 - 17:00)
Allowed on the exam: one A4 page (both sides) of *handwritten* notes.

The exam consists of 40 points distributed over 11 questions. Answers must be given in English or Swedish and should be clearly justified.

Grades: 3 : 20p, 4 : 26p, 5 : 34p

- Explain all solutions. A correctly explained solution with minor mistakes may render full points.
- Write clearly. Unreadable solutions will not be graded.
- Start each question on a new page and only write on one side of the page.
- Please make sure your exam is sorted with the answers in the natural order.
- Write down any assumptions you make.

Regular expressions and lexing (6p)

1) An identifier can contain letters, \$, _ and digits but must not start with a digit. Write a regular expression that matches identifiers. **(2p)**

2) Consider the following table describing a lexer: **(2+2p)**

regexp	*token*
$d(a\|b)^*d$	
aa	K
$a^+(a\|b)^*$	I
c	C

In the left column are the regular expressions and in the right column are the corresponding tokens that should be emitted. What sequence of tokens would be the result if the lexer is run on

1. *cadabdaa*
2. *aabaaca*

For full points you must explain why each token is generated in terms of the rules we have discussed in the course.

Grammars and parsing (14p)

3) Give an ambiguous grammar and show that it is ambiguous. **(2+2p)**

4) Consider a small expression language. **(4p)**

$$\begin{aligned} E &::= F \mid F+E \\ F &::= ID \mid NUM \end{aligned}$$

Extend the grammar in an unambiguous way with abitrary nested lists $[E,E,...]$. For example, the following expressions should be in your language.

- $[\,]$
- $[1,[x,y]]$
- $[[\,],[1],x]$

5) Does a shift/reduce conflict in an LR parser mean that the grammar is ambiguous? Explain why or why not. A simple yes/no answer gives no points. **(2p)**

6) Create a grammar that contains an $LR(0)$ shift/reduce conflict and show the existence of the conflict by constructing the $LR(0)$ automaton with the conflict marked. **(2+2p)**

Type checking (10p)

7) Consider the following expression language **(4p)**

$$E \;::=\; (E,E) \mid E+E \mid E == E \mid \texttt{true} \mid \texttt{false} \mid NUM \mid \texttt{fst}\ E \mid \texttt{snd}\ E$$

where (E,E) creates a pair, `fst` E takes a pair and returns the first element of the pair, and `snd` E takes a pair and returns the second element of the pair.

Given the following types

$$T \;::=\; (T,T) \mid \texttt{num} \mid \texttt{bool}$$

explain how to type check the language using a mix of natural language and pseudocode. NOTE: your type system must agree with the two examples in the next question.

8) Using your type system for the above language show that **(2+2p)**

1. $(true, 1+2)$ is type correct, and what type it has
2. $(1,2) == (1, true+2)$ is not type correct.

9) In the course we have discussed several different benefits with having a type checker in a compiler. Give two of those benefits. **(2p)**

Code generation. (10p)

10) Generate Trac42VM code for the following program. Be sure to follow the syntax of the source program closely. **(6p)**

```
int sum(int n, int a) {
  if (n==0) return a;
  return sum(n-1, a + n);
}
```

For full points mark clearly which part of the resulting Trac42VM code belongs to which part of the program.

11) Using psuedocode and natural language explain how to compile $E\ ?\ E\ :\ E$ to Trac42VM. **(4p)**