Exam in courses

TDDB 44 Compiler construction
TDDB 29 Compilers and interpreters
2003-12-16, 14.00 - 18.00

Aids: None.

Max = 32 points, 16 points to pass.

Teacher on duty: Jonas Wallgren

Problem 1 (4p) Formal languages and automata theory

NB! Only TDDB 29 (Compilers and Interpreters) students should solve this problem!

a) Give a DFA for the language L1 over {0,1} such that if a string conteins more than 3 zeroes then it must contain more than 2 ones.

b) Give a regular expression for the language L2 over {0,1} such that the number of ones in a string has the same parity as the number of xeroes (i.e. both odd or both even).

Problem 2 (4p) Symbol tables

Describe what happens in a hash based symbol table ad

a) declaration of a variable.

b) use of a variable.

c) entrance into a block

d) exit from a block.

Problem 3 (2p) Top-down parsing

The following grammar should be used for recursive descent parsing. What is/are the problem(s)? Rewrite the grammar to make it useable.

```
X -> aX | Xb | c
```

Problem 4 (4p) LR parsing

a) Give a context free grammar, for a language of your own choice, that is SLR(1) but not LR(0). Describe why it isn't LR(0).

b) Give a string, in the language, that need SLR(1) and show, using tables and stack, how it is parsed.

Problem 5 (4p): Intermediate code generation

Translate the following code segment to quadruples, postfix code, and abstract syntax tree:

```
while x<10 do
  if y<x
     then y:=y+1
     else x:=x+1;
```

Problem 6 (3p) Code optimization

a) What is a basic block? What is a loop?

b) Describe the loop optimization methods presented in the course, Use code examples.

Problem 7 (6p) Syntax directed translation

An Algol-like language is augmented with an `if2` statement in the following way:

```
<if2_statement> ::= if2(<expression_1>,<expression_2>)
                      none: <statement_1>
                       fst: <statement_2>
                       snd: <statement_3>
                      both: <statement_4>
                    endif2;
```

The `if2` statement works like the following nesting of `if` statements:

```
if <expression_1>
    then if <expression_2>
           then <statement_4>
           else <statement_2>
    else if <expression_2>
           then <statement_3>
           else <statement_1>;
```

Write the semantic rules - a syntax directed translation scheme - for translating the `cond3` statement to quadruples. Assume that the translation scheme is to be used in a bottom-up parsing environment using a semantic stack. Use the grammar rule above as a starting point, but maybe it has to be changed.

You are not allowed to define and use symbolic labels, i.e. all jumps should have absolute quadruple addresses as their destinations. Explain all the attributes, functions, and instructions that you introduce. State all your assumptions.

Problem 8 (3p) Memory management

What is done at

a) compilation of a subprogram declaration

b) compilation of a subprogram call

c) execution of a subprogram call

in a language with static memory management?

Problem 9 (2p) Boot strapping

Explain the concepts of rehosting and retargeting. Use T diagrams.

Problem 10 (4p) Code generation for RISC

NB! Only TDDB 44 (Compiler Construction) students should solve this problem!

a) What is branch prediction and when is it used? Give an example! Why is it important for pipelined processors?

b) Shortly explain software pipelining. Give a simple example.