Exam in course

TDDA 37, TDDB 44 Compiler construction 2000-12-20, 08.00 - 12.00

Aids: None.

Max = 32 points, 16 points to pass.

Jourhavande lärare: Jonas Wallgren (only by phone)
Problem 1 (2p) Phases and passes
What is a phase and what is a pass? What phases does a compiler normally consist of?

Problem 2 (2p) Symbol table
Describe how a symbol table in principle, independent of how it is represented, handles
a) Variable declaration,
b) Variable use,
c) Entrance into a block,
d) Exit from a block.

Problem 3 (4p) Top-down parsing
Explain and remedy the problems in the grammar

\[
X ::= \text{a}X | \text{b} | \text{a}Y\text{b} | p \\
Y ::= \text{b}Y | \text{Y}a | \text{b}X\text{a} | q
\]

which should be used for recursive descent parsing.

Problem 4 (4p) LR parsing
If the grammar

\[
X ::= \text{a}X | \text{b} | \text{a}Y\text{b} | p \\
Y ::= \text{b}Y | \text{Y}a | \text{b}X\text{a} | q
\]

where X is the start symbol, is SLR(1) or even LR(0) then show, using tables and stack, how
the string \text{ab}p\text{ab} is parsed. If the grammar is not then show how it could be rewritten to (at
least) SLR(1) (and still describe the same language).

Problem 5 (5p): Intermediate code generation
Translate the following code segment to quadruples, postfix code, and abstract syntax tree:

\[
\text{while } y>37 \text{ do} \\
\quad \text{if } i>15 \\
\quad \quad \text{then } x:=x+1 \\
\quad \quad \text{else } y:=y-1;
\]

Problem 6 (3p) Code optimization
What is a loop?
Explain, using clear examples, the loop optimization methods presented in the course.
Problem 7 (6p) Syntax directed translation

The following grammar rule describes a for loop:

```
<loop> ::= for <var> in <expr>..<expr> loop <stmt_list> end loop
```

<var> takes the values between the two expressions (the endpoints included) one by one, and for each value <stmt_list> is executed. Eg:

```
for x in 1..5 loop
  print(x);
end loop;
```

prints the numbers 1 to 5.

a) Write a syntax directed translation scheme, using attributes and semantic rules, for the grammar rule above. The values of the both <expr>s are computed in the beginning and cannot be changed during the execution of the loop. Neither can <var> be changed inside the loop.

b) What changes must be done to the solution to a) if <stmt_list> is allowed to change the loop variable and the endpoints? You don’t need to give a complete, detailed solution, just to state which parts and how, in principle.

Eg (b):
```
for x in a..b loop
  b:=b-1;  --changes upper endpoint
  x:=x+1;  --changes x (not the loop mechanism)
end loop;
```

Problem 8 (2p) Boot strapping

On the M machine there is a compiler for the high level language S. There is an executable version and a source code version - it’s written in S itself.

What is the most simple procedure to implement S on N, a machine where there is no S compiler?

Problem 9 (4p) Code generation for RISC

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.