LINKÖPINGS UNIVERSITET
Department of Computer Science
Jonas Wallgren/*

Exam in course

TDDA 37 Compiler construction 1999-12-17 kl 09.00 - 13.00

No books or other aids allowed.

Max = 32 points, 16 points needed to pass.

Teacher on duty: Jonas Wallgren

Problem 1 (2p) Phases and passes

Describe advantages and disadvantages of multi-pass compiling.

Problem 2 (3p) Symbol table

Describe advantages and disadvantages of the symbol table representations presented in the course.

Problem 3 (4p) Top-down parsing

Describe the problems that could exist in a context free grammar that should be used for top-down parsing. Also describe how the problems could be solved. Does the solution always work?

Problem 4 (4p) LR parsing

Explain LR parsing. Don't primarily describe how things are done, but why they are done.
(I.e. not primarily how technical constructions s.a. items, automata, and tables work, but the motivs behind all that technology.)

Problem 5 (5p): Intermediate code generation

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

```
for i:=1 to 20 do
  if i>15
    then x:=x+1
    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 (5p) Syntax directed translation

To a Pascal-like programming language a `restartblock` statement is added according to the following grammar fragment:

<block> ::= `begin` <stmt_list> `end`
<stmt_list> ::= <stmt_list> <stmt> | ε
<stmt> ::= <assignment> | ... | `restartblock`

(where "..." represents all other kinds of statements.)

`restartblock` means that the execution restarts at the closest enclosing block.
Example:

```
begin
    i:=7;
L1: begin
        j:=j+1;
        if j<i
L2:         then restartblock
            else i:=i+1
    end;
end;
```

`restartblock` in `L2` thus leads to a jump back to `L1`.

a) Write a syntax directed translation scheme, with attributes and semantic rules, for the grammar fragment above.

c) Which problems would arise in the translation scheme handling if instead of `restartblock` there was an `exitblock` that jumped to the `end` of the closest enclosing block (instead of its `begin`)?

Problem 8 (2p) Bootstrapping

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

What is the simplest way to implement S on N, a machine where no S compiler exists?

Problem 9 (4p) Code generation for RISC

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

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