

Tentamen i kurs

TDDA 37 Kompilatorkonstruktion 1999-12-17 kl 09.00 - 13.00

Hjälpmedel: Inga.

Poänggränser: Max = 32 poäng, för godkänd krävs 16 poäng.

Jourhavande lärare: Jonas Wallgren

Uppgift 1 (2p) Faser och pass

Beskriv fördelar och nackdelar med flerpasskompilering.

Uppgift 2 (3p) Symboltabell

Beskriv fördelar och nackdelar med de i kursen presenterade symboltabell-representationerna?

Uppgift 3 (4p) Top-downparsning

Ange de problem som kan finnas i en kontextfri grammatik som skall användas för top-downparsning. Ange också hur problemen kan lösas och om lösningsmetoderna alltid fungerar.

Uppgift 4 (4p) LR-parsning

Förklara LR-parsning. Beskriv inte i första hand hur man gör, utan varför man gör det. (Alltså inte i första hand hur tekniska konstruktioner ss items, automater och tabeller fungerar, utan motiven bakom all denna teknik.)

Uppgift 5 (5p): Mellankodsgenerering

Överför följande kodavsnitt till kvadrupler, postfixkod, och abstrakt syntaxträd:

```
for i:=1 to 20 do
  if i>15
    then x:=x+1
    else y:=y-1;
```

Uppgift 6 (3p) Kodoptimering

Vad är en loop?

Förklara, med tydliga exempel, de i kursen presenterade loopoptimeringsmetoderna.

Uppgift 7 (5p) Syntaxstyrd översättning

Ett Pascal-liknande språk utökas med en `restartblock`-sats enligt följande grammatikavsnitt:

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

(där "..." står för alla andra möjliga typer av satser)

`restartblock` innebär att exekveringen börjar om på det närmast omgivande blocket.

Exempel:

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

`restartblock` i L2 innebär alltså ett återhopp till L1.

a) Skriv ett syntaxstyrt översättningsschema, med attribut och semantiska regler, för grammatikavsnittet ovan.

c) Vad skulle det uppstå för problem i hanteringen av översättningsschemat om istf `restartblock` fanns `exitblock` som hoppade till närmast omgivande blocks `end` (istf `begin`)?

Uppgift 8 (2p) Bootstrapping

På maskinen M finns en kompilator för högnivåspråket S. Den finns både i en exekverbar form och som källkod - den är skriven i S självt.

Hur gör man för att enklast implementera S på N, en maskin där ingen S-kompilator finns?

Uppgift 9 (4p) Kodgenerering för RISC

a) Vad är branchprediktion och när används det? Ge exempel! Varför är det viktigt för pipelineade processorer?

b) Förklara kort software pipelining. Visa med ett enkelt exempel.