Tentamen i kurs

TDDA 37 Kompilatorkonstruktion 1999-04-17 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

Varför kan det behövas flera pass i en kompilator?
Pascal utformades för enpasskompilering. Varför kan det vara önskvärt?

Uppgift 2 (3p) Symboltabell

a)Beskriv hur den i kursen presenterade hashbaserade symboltabellsmodellen hanterar
1) Påbörjande av ett nytt block.
2) Avslutande av ett block.

b) Vilken är den största nackdelen med en linjär lista som symboltabellrepresentation?

Uppgift 3 (4p) Top-downparsning

Följande grammatik, där A är startsymbol, skall användas för top-downparsning.

```
A::=Ax|By|p
B::=Ay|Bx|q
```

Om det inte är några problem med grammatiken: Skriv en parser. Variabler behöver inte deklareras. Antag att det finns en procedur `scan()` som uppdaterar den globala variabeln `token`.
Om det är problem med grammatiken: Red ut problemen och deras lösning.

Uppgift 4 (5p) LR-parsning

a) Konstruera LR-item-mängderna för följande grammatik, där N är startsymbol:

```
N::=NEENx|0
E::=ENNEx|1
```

b) Avgör, huvudsakligen mha konstruktionen i a) om följande grammatik är LR(0):

```
N::=NEEN|0
E::=ENNE|1
```

c) Visa, mha parsetabeller och stack, hur strängen `0110x110x` parsas (enligt grammatiken i a)

Uppgift 5 (5p): Mellankodsgenerering

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

```
while y<20 do
  if x>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 (4p) Syntaxstyrd översättning

En enkel variant av en `FOR`-sats kan beskrivas med regeln

```
<for-stat> ::= FOR i := <expr>1 TO <expr>2 DO <S>
```

Betydelsemässigt är ovan beskrivna sats ekvivalent med följande:

```
BEGIN
   i := <expr>1;
   temp :=<expr>2;
   WHILE i <= temp DO
   BEGIN
       <S>;
       i := i + 1;
   END;
END;
```

Skriv ett syntax-styrt översättningsschema med attribut och semantiska regler, för översättning av `FOR`-satsen till kvadrupler. Antag att översättningsschemat ska implementeras i bottom-up-parsningmiljö. Förklara införda attribut och funktioner. Låt `<expr>`$_1$, `<expr>`$_2$ och `<S>` vara icke-terminala symboler som ni inte behöver generera kvadrupler för, och antag att resultatet av t.ex. `<expr>` finns tillgängligt i attributet `<expr>.ADDR`.

Uppgift 8 (2p) Bootstrapping

Förklara begreppen rehosting och retargeting. Beskriv hur de utförs. Använd gärna T-diagram.

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