Örebro universitet
Institutionen för teknik
Thomas Padron-McCarthy (Thomas.Padron-McCarthy@tech.oru.se)

Tentamen i

Kompilatorer och interpretatorer

för Dataingenjörsprogrammet m fl

fredag 24 augusti 2007 kl 08:00 - 12:00 i L001

Gäller som tentamen för:
TDK104 Kompilatorer och interpretatorer, provkod 0100

Hjälpmedel:	Inga hjälpmedel.
Poängkrav:	Maximal poäng är 30. För betyget 3 krävs 15 poäng.
Resultat:	Meddelas på kursens hemsida senast måndag 10 september 2007.
Visning och frågestund:	Tisdag 11 september 2007 kl 12:00-12:30 i mitt rum (T2220). Därefter kan tentorna hämtas på expeditionen.
Examinator och jourhavande:	Thomas Padron-McCarthy, telefon 070-7347013.

- Skriv tydligt och klart. Lösningar som inte går att läsa kan naturligtvis inte ge några poäng. Oklara och tvetydiga formuleringar kommer att misstolkas.
- Skriv den personliga tentamenskoden på varje inlämnat blad. Skriv *inte* namn eller personnummer på bladen.
- Skriv bara på en sida av papperet. Använd inte röd skrift.
- Antaganden utöver de som står i uppgifterna måste anges.
- Skriv gärna förklaringar om hur du tänkt. Även ett svar som är fel kan ge poäng, om det finns med en förklaring som visar att huvudtankarna var rätt.

LYCKA TILL!

Uppgift 1: Grammatiker (5p)

Ett språk innehåller terminalerna **a**, **b**, **c**, **d** och **e**. Grammatiken för språket innehåller icke-terminalerna *S*, *T* och *U*, och följande produktioner:

```
S -> a T | b U
T -> b c U e
U -> c d
```

Startsymbolen är *S*.

(a) (2p)

Ange vilket språk grammatiken beskriver, genom att räkna upp alla strängar (dvs sekvenser av terminaler) som kan uttryckas i språket.

Visa hur du fick fram svaret.

(b) (1p)

I ett annat språk ska man kunna skriva godtyckligt många förekomster av strängen **a a**, följt av strängen **bc**. Med "godtyckligt många" menas noll, en eller flera. Exempel på strängar som ska accepteras:

- **b c**
- **a a b c**
- **a a a a a a a a b c**

Skriv en grammatik för detta språk. Glöm inte att ange vad som är startsymbol. Grammatiken ska vara entydig.

(c) (1p)

I deluppgiften ovan stod det att grammatiken skulle vara entydig. Det är motsatsen till att grammatiken är tvetydig. Förklara med exempel vad som menas med att en grammatik är entydig respektive tvetydig.

(d) (1p)

Skriv ett reguljärt uttryck som beskriver språket i deluppgift a.

Uppgift 2: Kompilatorer (5 p)

En av faserna som en kompilator brukar delas in i är **optimering**.

a) Vilka andra faser finns det, och i vilken ordning kommer de?

b) I en nyare upplaga av "drakboken" har man delat in optimeringsfasen i två olika faser: **maskinoberoende optimering** och **maskinberoende optimering**. Varför har man gjort den uppdelningen?

Uppgift 3: Parsning (5 p)

a)

Gör eventuella transformationer av grammatiken i deluppgift 1 a som krävs för att den ska lämpa sig för implementation av en prediktiv recursive-descent-parser (engelska: predictive recursive-descent parser).

b)

Skriv den prediktiva recursive-descent-parsern i ett språk som åtminstone liknar C, Java eller Pascal. Du behöver inte skriva exakt korrekt programkod, men det ska framgå vilka procedurer som finns, hur de anropar varandra, och vilka jämförelser med tokentyper som görs. Du kan anta att det finns en funktion som heter **scan**, och att den returnerar typen på nästa token.

Uppgift 4: Syntaxstyrd översättning (5 p)

a)

Skriv en grammatikregel för if-satsen i C.

b)

Skapa ett syntaxstyrt översättningsschema, med produktioner och semantiska aktioner, för översättning av while-satsen till mellankod. Välj själv om du vill generera ett abstrakt syntaxträd, postfixkod för en stackmaskin eller treadresskod.

Uppgift 5: Mellankod (5 p)

```
while (a + (b + c) < a + b + c) {
  if (i > j) {
    a = 200;
  }
  else {
    c = b * c + a * j;
    j = j + 1;
  }
  a = j + 1;
}
```

Översätt ovanstående programavsnitt till *två* av följande tre typer av mellankod.

a) ett abstrakt syntaxträd (genom att rita upp trädet!)

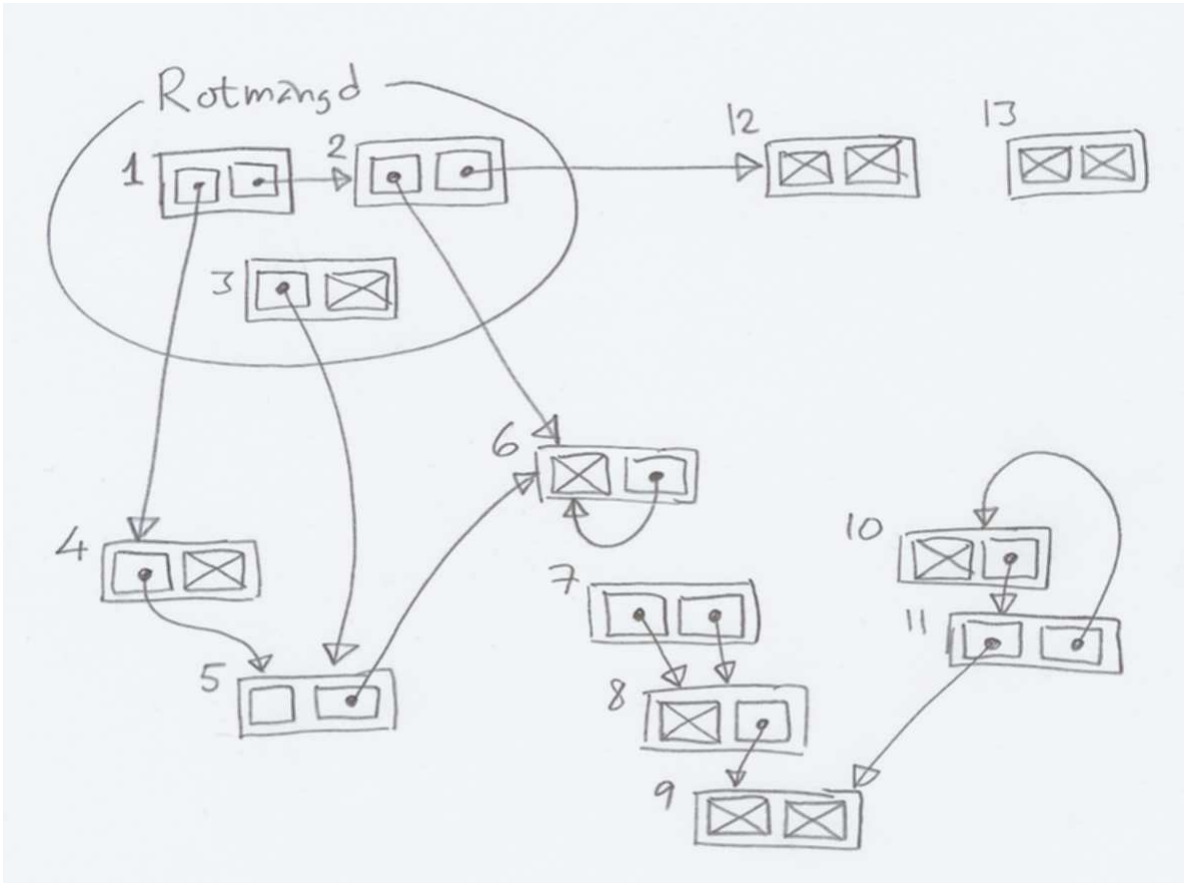
b) postfixkod för en stackmaskin

c) treadresskod

Observera: Det finns tre deluppgifter i uppgiften ovan. Välj ut och besvara (högst) *två* av dessa. (Skulle du svara på alla tre, räknas den med högst poäng bort.)

Uppgift 6: Run-time-omgivningar (5 p)

Antag att minnet innehåller dessa sju dataobjekt:



a) Antag att vi använder skräpsamlingsmetoden "mark and sweep". Vilka av objekten kommer att markeras i markeringsfasen?

b) Vilka av objekten kommer att tas bort i sopfasen?

c) Antag att vi i stället använder referensräkning. Kan situationen som visas på bilden uppstå? (Vi antar att vi inte är mitt i en uppstädning.) Motivera svaret.

d) Vi använder fortfarande referensräkning. Ange värdet på referensräknaren för vart och ett av dataobjekten.

e) Nu använder vi mark and sweep. Vi sätter de tre pekarna som pekar ut ur rotmängden från dataobjekt 1 och 2 till NULL. Beskriv vad som händer vid nästa skräpsamling.