

Örebro universitet
Institutionen för naturvetenskap och teknik
[Thomas Padron-McCarthy \(thomas.padron-mccarthy@oru.se\)](mailto:thomas.padron-mccarthy@oru.se)

Tentamen i

Kompilatorer och interpretatorer

för Dataingenjörsprogrammet m fl

onsdag 3 januari 2018

Gäller som tentamen för:

DT125G Kompilatorer och interpretatorer, provkod 0100
DT3030 Datateknik C, Kompilatorer och interpretatorer, provkod 0100

Hjälpmedel:	Inga hjälpmedel.
Poängkrav:	Maximal poäng är 37 För godkänt betyg krävs totalt minst 21 poäng, varav minst 8 poäng på uppgift 1.
Resultat:	Meddelas på kursens hemsida eller via e-post senast onsdag 24 januari 2018.
Återlämning av tentor:	Elektroniskt via Studentforum.
Examinator och jourhavande:	Thomas Padron-McCarthy, telefon 070 - 73 47 013.

- 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!

Formelsamling

1. Eliminering av vänsterrekursion

En vänsterrekursiv grammatik kan skrivas om så att den inte är vänsterrekursiv. Antag att en regel (eller, korrektare uttryckt, två produktioner) i grammatiken ser ut så här:

$$A \rightarrow A x \mid y$$

A är en icke-terminal, men **x** och **y** står för godtyckliga konstruktioner som består av terminaler och icke-terminaler.

Regeln ersätts av följande två regler (eller, korrektare uttryckt, tre produktioner), som beskriver samma språk men som inte är vänsterrekursiva:

$$\begin{aligned} A &\rightarrow y R \\ R &\rightarrow x R \mid \text{empty} \end{aligned}$$

2. Vänsterfaktorisering

Antag att grammatiken innehåller denna regel (två produktioner):

$$A \rightarrow x y \mid x z$$

A är en icke-terminal, men **x**, **y** och **z** står för godtyckliga konstruktioner som består av terminaler och icke-terminaler.

Skriv om till dessa tre produktioner:

$$\begin{aligned} A &\rightarrow x R \\ R &\rightarrow y \mid z \end{aligned}$$

Uppgift 1 (10 p)

En kompilators arbete brukar delas in i ett antal faser. Ange vilka det är, och förklara kort vad varje fas gör. Vad är in- och utdata från respektive fas?

Uppgift 2 (5 p)

Det här är ett programavsnitt i ett C-liknande språk:

```
x = 10;
y = 20;
z = 30;
while (x != y) {
    if (x - y - z < 0)
        x = (x + y) / 2;
    z = z + 1;
}
```

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

- ett abstrakt syntaxträd (genom att rita upp trädet!)
- postfixkod för en stackmaskin
- 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 3 (5 p)

Förklara, gärna med exempel, följande termer från kompilatorområdet. Det kan vara lämpligt med några få meningar för varje term.

- Looputrullning
- Aktiveringspost
- Tvetydig grammatik
- FIRST()-konflikt
- Kvadrupler (för lagring av treadresskod)

Scenario till uppgift 4-7

Som vi vet var det Donald Trump som vann presidentvalet i USA förra året. Han besegrade Hillary Clinton.

Media, och anhängare till Hillary Clinton, talar mycket om att det förekommit hemligt samarbete mellan Ryssland och personer som deltog i Donald Trumps valkampanj. Anhängare till Donald Trump talar hellre om samarbetet mellan Ryssland och personer som deltog i Hillary Clintons valkampanj.

Nu ska den amerikanska federala polisen, FBI, försöka utreda alltihop. De ska göra ett program där man matar in personer och vilka personer som arbetar för andra personer. Inmatningen kan se ut så här:

```
person 1 "Vladimir Putin"  
person 2 "Donald Trump"  
person 3 "Hillary Clinton"  
person 4 "Sergey Kislyak"  
4 arbetar för 1  
person 7 "John Podesta"  
person 5 "Paul Manafort"  
5 arbetar för 2  
person 6 "Carter Page"  
6 arbetar för 2  
7 arbetar för 3  
7 arbetar för 1  
klart
```

Inmatningsspråket består av "person-kommandon", som anger att en person finns och vilket nummer hon har, "arbetar för"-kommandon, som anger att en viss person arbetar för en annan person, och "klart"-kommandot, som avslutar inmatningen.

Uppgift 4 (2 p)

Vilka terminaler, dvs typer av tokens, behövs för att man ska kunna skriva en grammatik för inmatningsspråket i scenariot?

Uppgift 5 (4 p)

Skriv en grammatik för ett inmatningsspråket. Startsymbolen ska vara **inmatning**, som representerar en komplett inmatning enligt scenariot ovan.

Uppgift 6 (3 p)

a) Ett parse-träd (ibland kallat "konkret syntaxträd") innehåller noder för alla icke-terminaler. Rita upp parse-trädet för det här köpet, enligt din grammatik i uppgiften ovan:

```
person 1 "Vladimir Putin"  
3 arbetar för 1  
klart
```

b) I exemplet ovan matar man in att person 3 arbetar för person 1. Men det finns ingen person 3 i inmatningen. Förklara hur det påverkar parse-trädet!

Uppgift 7 (8 p)

Skriv en prediktiv recursive-descent-parser för språket i scenariot, i ett programspråk som åtminstone liknar C, C++, C# eller Java. 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**, som returnerar typen på nästa token, och en funktion som heter **error**, som man kan anropa när något gått fel och som skriver ut ett felmeddelande och avslutar programmet.
