

Kompilatorer och interpretatorer: Hemtentamen 2021-08-27

Det här är hemtentan som går fredag 27 augusti 2021 i kursen DT135G Kompilatorer och interpretatorer, provkod A001. Ansvarig lärare är [Thomas Padron-McCarthy](mailto:thomas.padron-mccarthy@oru.se) (thomas.padron-mccarthy@oru.se), telefon **070-73 47 013**.

Tid: 08:15 - 13:15

Instruktioner

1. Den här hemtentan ersätter den planerade salstentan.
2. Tentatiden är utökad med en extra timme för att täcka in problem med e-post, inscanning eller fotografering av diagram, och liknande.
3. Uppgifterna ska lösas enskilt, dvs inga grupper av två eller flera studenter.
4. **Du får använda dator, böcker och vilka andra hjälpmedel som helst**, men du får inte samarbeta eller fråga någon (utom mig). Exempelvis är det tillåtet att söka och läsa på webbplatser som Stack Overflow, men inte att ställa egna frågor. Du kan alltså provköra kommandon och program, om du vill.
5. **Diskutera inte uppgifterna eller dela med dig av svar förrän tidigast dagen efter tentan.**
6. Lös de angivna uppgifterna och samla svaren på lämpligt sätt, till exempel i en PDF. Skicka sen in lösningarna till mig i [Blackboard](#). (Gå till kursens sida i Blackboard, gå in i mappen **Hemtentor**, och välj den rätta tentan. Där kan man sedan välja att skicka in sin lösning. Då blir bedömningen anonym.)
7. Om det skulle bli problem med Blackboard, kan man i nödfall skicka in svaren antingen som ett kursmeddelande i Blackboard eller via vanlig e-post (thomas.padron-mccarthy@oru.se), senast vid tentatidens slut. Då blir bedömningen inte anonym.
8. I Blackboard ser du att du skickat in din lösning. Om du i stället skickade in med e-post, och inte senast en timme efter tentatidens slut fått ett svar från mig med en bekräftelse på att du skickat in svaren, bör du kontakta mig, enklast genom att ringa eller SMS:a mig (ifall det är e-posten som inte fungerar). Tänk på att en del mailtjänster (särskilt Microsoft-tjänster som Hotmail.com, Outlook.com och Live.com) ibland kastar bort brev med bilagor, utan att meddela det.
9. Skriv gärna svaren i ett ordbehandlingsprogram. Rita gärna eventuella diagram i ett ritprogram. Det är inte förbjudet att skriva och rita för hand, men då måste text och bilder scannas in eller fotograferas. Det finns scanner-appar till Android och iPhone, till exempel Adobe Scan, som ger bättre resultat än att bara ta vanliga kort med kameran.
10. Om du behöver fråga något, så kontakta gärna mig. Ring eller skicka SMS, för jag kanske inte kommer att sitta vid datorn hela tiden.
11. Oklara och tvetydiga formuleringar kommer att misstolkas. Lösningar som inte går att läsa eller förstå kan naturligtvis inte ge några poäng.
12. Antaganden utöver de som står i uppgifterna måste anges. Gjorda antaganden får inte förändra den givna uppgiften.
13. 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.

14. Maximal poäng är 33. För godkänt betyg krävs minst 19 poäng.
15. Resultat meddelas senast 15 arbetsdagar efter tentamensdatum. Eftersom svaren skickas in elektroniskt scannas tentorna inte för retur.

Information från institutionen angående fusk

Instruktioner inför digital hemtentamen/ examination

Jag vill undvika fusk - hur gör jag? / All form av fusk anmäls

Du ska följa instruktionerna för uppgiften. Om du är osäker, fråga ansvarig lärare om något i instruktionerna är oklart.

Du får inte samarbeta. Det här är en individuell examinationsuppgift. Du ska inte prata med någon, ställa frågor till eller ta hjälp av andra studenter eller kurskamrater. Att hjälpa andra under en individuell examination är också fusk.

Lägg undan mobilen. Stäng av sociala medier.

Du får inte använda hjälpmedel. Det vill säga att du får inte använda dig av något annat än det som står angivet i instruktionerna.

Du får inte använda andra formuleringar än dina egna. Dina svar ska vara **självständigt formulerade** och **redovisa dina kunskaper**. Det betyder att inga citat eller referat ska förekomma i dina svar om det inte står i instruktionerna att du får använda citat eller referat.

Dina svar kontrolleras via Urkund.

All misstanke om fusk anmäls till universitetets rektor och kan leda till en prövning i universitets disciplinnämnd.

Konsekvenser av fusk

Om du fuskar kan detta leda till en avstängning som kan få följder både för dina studier och privat:

- Uteblivet studiemedel som t.ex. som kan påverka din möjlighet att behålla din bostad.
- Ingen åtkomst till digitala plattformar.
- Du kan behöva meddela dina kursare om att du är fälld för fusk, om du t.ex. ingår i ett grupparbete på en pågående kurs men blir avstängd.
- Tillfällen för examination går förlorade vilket kan innebära att du inte kommer vidare i dina studier nästa läsperiod/termin och din studiegång blir därmed påverkad.
- Beslutet är en offentlig handling som begärs regelbundet ut av en nyhetsbyrå. Och kan även begäras ut av framtida arbetsgivare eller andra.
- Om du fuskar dig igenom din utbildning har du inte den kunskap som arbetsmarknaden förväntar av dig.

OBS: Tänk efter en gång till innan du påbörjar och genomför din tentamen!

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:

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

Scenario till uppgifterna

Vaksamma Valdemars Vakt AB tillhandahåller väktare, som vaktar olika lagerlokaler och liknande ställen. En väktare är, skriver [Wikipedia](#), "en person som är anställd av ett bevakningsföretag för att utföra olika bevakningssysslor såsom yttre samt inre tillsyn i och runt objekt (så kallad rondering), stationär bevakning på objekt, områdesbevakning i förutbestämda områden samt rycka ut och åtgärda utlösta larm på både privata och företagsobjekt."

Vaksamma Valdemars Vakt AB har specialiserat sig på stationär bevakning, där en eller flera väktare sitter och vaktar en plats:



(Foto av Brad & Sabrina, <https://www.flickr.com/photos/19078064@N00/184665954>. Licens: [Creative Commons Attribution 2.0 Generic](#).)

Vi ska skapa ett inmatningspråk för att ange väktare, platser och vilka väktare som vaktar vilken plats. Språket består av fyra olika kommandon:

- Kommandot **väktare**, som anger att det finns en väktare med ett visst namn. Kommandot består av ordet **väktare**, följt av ett namn inom citationstecken, och avslutas med ett semikolon. Exempel: **väktare "Bengt Bengtsson";**
- Kommandot **plats**, som anger en plats som ska vaktas. Kommandot består av ordet **plats**, följt av ett namn inom citationstecken, och avslutas med ett semikolon. Exempel: **plats "Bromma flygplats";**
- Kommandot **vaktar**, som anger att en väktare vaktar en plats. Man skriver namnet på en väktare, följt av ordet **vaktar**, följt av namnet på en plats, och kommandot avslutas med ett semikolon. Exempel: **"Bengt Bengtsson" vaktar "Bromma flygplats";**
- Kommandot **klart**, som avslutar inmatningen. Exempel: **klart;**

Exempel på en komplett inmatning:

```
vaktare "Bengt Bengtsson";  
plats "Bromma flygplats";  
vaktare "Cecilia Chowdhury";  
vaktare "Donald D. Duck";  
"Bengt Bengtsson" vaktar "Bromma flygplats";  
plats "Tybble centrum";  
"Cecilia Chowdhury" vaktar "Tybble centrum";  
klart;
```

Kommandon kan skrivas på fritt format, dvs att man kan stoppa in godtyckliga mellanslag och radbrytningar, utom inuti namnen som avgränsas med citationstecken.

Uppgift 1 (4 p)

- a) Vilka terminaler behövs för att man ska kunna skriva en grammatik för språket?
- b) Det kan hända att en eller flera av terminalerna inte har fixa lexem, utan kan se ut på olika sätt. Skriv i så fall reguljära uttryck som beskriver hur de får se ut.

Uppgift 2 (4 p)

Skriv en scanner som kan dela upp inmatningen i tokens. Du får själv välja om du ska använda Flex eller skriva den för hand i C, C++ eller ett liknande språk. Scannern ska bestå av en funktion som returnerar en token-kod. Du kan anta att token-koderna finns definierade i form av makron eller enum-konstanter, ungefär som Bison gör när man har en Bison-grammatik med **%token**-deklarationer.

Uppgift 3 (5 p)

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

Uppgift 4 (4 p)

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

```
väktare "Hjalmar";  
väktare "Hulda";  
"Hjalmar" vaktar "Hulda";  
klart;
```

Uppgift 5 (3 p)

I uppgiften ovan står det att väktaren Hjalmar vaktar Hulda, men Hulda är ju ingen plats, utan en annan väktare! Hur påverkar det:

- a) scannern
b) parsern
c) andra delar av programmet

Uppgift 6 (8 p)

Skriv en prediktiv recursive-descent-parser för språket. i ett språk 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. Du kan anta att token-koderna finns definierade i form av makron eller enum-konstanter.

Om grammatiken från uppgift 3 ovan inte lämpar sig för en prediktiv recursive-descent-parser, måste den först göras om.

Uppgift 7 (5 p)

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

```
r = 0;
s = 0;
while (r < 100) {
    if (r % 2 == 0)
        r = r + 3;
    else
        r = r + 5;
    s = s + r;
}
```

Översätt ovanstående programavsnitt till *två* av följande tre typer av mellankod. (Skulle du svara på alla tre, räknas den med högst poäng bort.)

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