

Tenta i Programmeringsparadigm - DD1362 2019-08-16 08.00-13.00

Inga hjälpmedel är tillåtna. Skriv inga svar på blanketten. För godkänt på en del krävs utförliga svar med kompletta meningar på samtliga (del)uppgifter för den delen. Ett grovt fel ger underkänt. En del blir godkänd även med ett halvgrovt fel. Varje del bedöms oberoende av resultat på de andra delarna. Eventuella bonuspoäng gäller inte längre. Lycka till önskar Marcus och Per!

Funktionell programmering

1. Funktionen `sorted` i Python tar in en lista och returnerar en sorterad kopia. Om man vill att en lista med strängar ska sorteras på längd så kan man skicka med en funktion som används som jämförelse. Exempel:

```
>>> v = ["Sophia", "Marcus", "Anna", "Gunnar"]
>>> sorted(v, key=len)
['Anna', 'Sophia', 'Marcus', 'Gunnar']
>>> sorted(v)
['Anna', 'Gunnar', 'Marcus', 'Sophia']
>>> sorted(v, key=lambda x:x[-1]=='a')
['Marcus', 'Gunnar', 'Sophia', 'Anna']
>>>
```

- (a) Vad kallas en funktion som tar en funktion som parameter?
- (b) Vilken notation används för anonyma funktioner i Haskell?

2. Studera följande Haskellkod:

```
f :: Integer -> Integer
f n
  | n == 0 = 1
  | otherwise = n * f (n-1)

g :: Integer -> Integer
g n = h 1 n

h :: Integer -> Integer -> Integer
h a 0 = a
h a n = h (a*n) (n-1)
```

Vilken eller vilka av f , g och h i programmet ovan är svansrekursiv? Motivera ditt svar noggrant med kursens terminologi.

3. En av Haskell's inbyggda funktioner heter `filter`. Du behöver inte skriva dess parametrar i rätt ordning i någon av uppgifterna, men du behöver veta vilka de är.

- (a) Vad är typssignaturen för `filter`?
- (b) Beskriv i detalj vad `filter` gör.

En uppgift till på nästa sida.

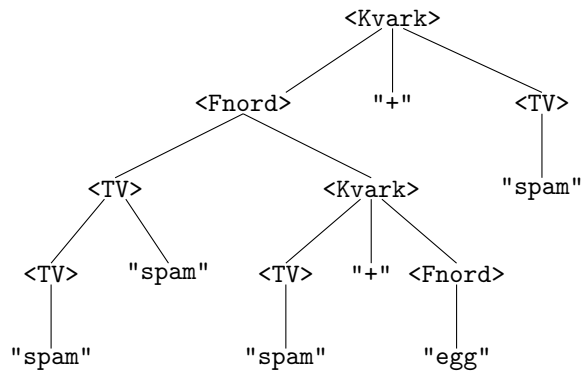
4. En föreläsning handlade om olika sorters polymorfism. För var och en av dessa olika sorters polymorfism, namnge vilken sort det är på formen "X polymorfism". Förklara i detalj varför exemplet beskriver just denna sorts polymorfism.
- (a) Ett verksamhetssystem är programmerat i ett objektorienterat språk. En superklass heter **Anställd** som innehåller metoden **agera**. Subklasserna till **Anställd** heter **Programmerare**, **Chef** och **Projektledare**. Varje subklass till **Anställd** har en egen metod **agera** som fungerar annorlunda än den i superklassen **Anställd**. Programmet kan innehålla en lista över flera objekt som är instanser av subklasser till **Anställd** och anropa metoden **agera** på dessa.
 - (b) Haskells inbyggda funktion `map` kan ta in såväl strängar som listor av heltal som en av sina parametrar.
 - (c) Additionsoperatoren kan överlagras i vissa språk så att `1+1 == 2` och `"1"+"1" == "11"` fungerar.
 - (d) I vissa språk kan det förekomma två funktioner med samma namn men olika parameteruppsättning. Vid funktionsanrop väljer kompilatorn vilken av dem som ska anropas.
-

Formella språk och syntaxanalys

5. Vad är definitionen av ett *formellt språk* (inom datavetenskap/matematik)?
6. Följande meningar använder terminologin från kursavsnittet, men på ett konstigt och/eller felaktigt sätt som indikerar att personen som yttrat meningarna inte riktigt vet vad hen pratar om. För var och en av meningarna, förklara vad som är konstigt.
- För de av er som är mer bekväma med kursens terminologi på engelska ges även engelska översättningar, du kan själv välja om du vill utgå från de engelska eller svenska versionerna i dina svar.
- (a) "Den lexikala analysen visade att strängen 'aabaab' var syntaktiskt korrekt."
(Engelska: "The lexical analysis showed that the string 'aabaab' was syntactically correct.")
 - (b) "Eftersom språket är reguljärt kan vi köra det genom den ändliga automaten."
(Engelska: "Since the language is regular we can run it through the finite automaton.")
 - (c) "Eftersom automaten har en loop så följer det från pumping-lemmat att språket är oändligt stort."
(Engelska: "Since the automaton has a loop it follows from the pumping lemma that the language is infinite.")

En uppgift till på nästa sida.

7. Betrakta följande syntaxträd som erhållits genom att parsas någon sträng x enligt någon grammatik.



- (a) Vad är strängen x som parsats?
- (b) Hur ser grammatiken ut som använts vid parsningen av x ? (Du behöver bara beskriva de delar av grammatiken som går att utläsa från syntaxträdet.)
- (c) Är språket som grammatiken beskriver oändligt? Det finns tre möjliga svar på denna fråga: "ja", "nej", och "det finns inte tillräckligt mycket information för att avgöra detta".

Paradigmdelen på sista sidan.

Paradigm

8. Tänk dig en delmängd av Python med variabler, tilldelningar, listor, egendefinierbara funktioner, if, for, while, tilldelningar och uttryck. Detta är en delmängd av egenskaper som många språk har och vi vill se vilket/vilka paradigm och språkegenskaper som denna delmängd använder.
 - (a) Räkna upp minst 3 språkegenskaper som är centrala för objektorienterad programmering. Förklara för var och en av dessa om de är implementerade i den beskrivna delmängden ovan.
 - (b) Implementerar denna delmängd strukturerad programmering? Vilka egenskaper finns och vilka finns inte?
9. Namnge och beskriv följande språkegenskaper i detalj.
 - (a) Namnge egenskapen att ett uttryck var som helst kan ersättas med sitt värde.
 - (b) Beskriv i detalj tanken bakom föregående egenskap och beskriv en språkfeature (till exempel en typ av sats) som är omöjlig att kombinera med föregående egenskap.
 - (c) Namnge egenskapen att ett och samma namn, exempelvis *y*, samtidigt kan ha olika värden i två olika delar av programmet *utan att något värde har förändrats, skrivits över eller raderats* och att på en tredje plats kan *y* vara odefinierat.
 - (d) Beskriv i detalj tanken bakom föregående egenskap.
10. På föreläsningarna visades korutiner. Beskriv i detalj vad som gör korutiner speciella med såpass mycket detaljnivå att det inte kan blandas ihop med någon annan typ av funktion eller språkegenskap som vi har pratat om i kursen.
11. Turingmaskinen kan växla mellan olika tillstånd under sin körning. Beskriv vilken eller vilka faktorer som påverkar vad som blir nästa tillstånd.