

Tenta i Programmeringsparadigm 2017-01-13 14.00-17.00

Inga hjälpmedel är tillåtna. Skriv inga svar på blanketten. Bonuspoäng från Inet-labben hösten 2016 kommer automatiskt att tillgodoräknas på den del av tentan där de gör mest nytta. Varje del är på 20 poäng. För att bli godkänd på en del krävs 12 poäng på den delen. **Du behöver inte skriva de delar där du redan är godkänd på kontrollskrivningen.**

Del 1: Funktionell programmering

1. Två strikt växande talserier (listor) av heltal $R = [r_0, r_1, \dots]$ och $S = [s_0, s_1, \dots]$ följer följande regler:

- $r_0 = 1$
- $s_0 = 2$
- $r_n = r_{n-1} + s_{n-1}$
- S innehåller alla positiva tal som inte förekommer i R .

De första talen i sekvenserna är

$$R = [1, 3, 7, 12, 18, 26, 35, 45, 56, 69, 83, 98, 114, 131, \dots]$$

$$S = [2, 4, 5, 6, 8, 9, 10, 11, 13, 14, 15, 16, 17, 19, \dots]$$

- (a) Skriv en funktion i Haskell som givet ett heltal $n \geq 1$ beräknar och returnerar en lista med de första n talen i R . Du får använda hjälpfunktioner. (6p)

Krav:

- Ange typsignatur för alla funktioner du definierar.
 - Du behöver använda rekursion som en central del av din lösning.
 - Alla rekursiva funktioner du definierar måste vara svansrekursiva.
- (b) Lista de funktionsanrop som görs, i rätt ordning med rätt parametrar, då din huvudfunktion anropas för att returnera en lista med de 3 första elementen i R . För att få poäng här behöver alla dina funktioner uppfylla kravlistan i (a) och åtminstone vara i närheten av att returnera rätt svar. (2p)
2. (a) Tigist har implementerat en funktion som returnerar en oändlig lista. Den har typsignaturen `hofstadterSequence :: [Integer]`. Skriv en funktion som tar en `Int` som parameter, anropar Tigists funktion och returnerar det n :te talet. Du får inte använda rekursion. (2p)
- (b) Namnge den egenskap hos Haskell som (bland annat) gör det möjligt att anropa en funktion som returnerar en oändligt lång lista utan att själv köras för evigt. (1p)
3. Skriv en Haskellfunktion som givet ett positivt heltal n returnerar en lista med alla positiva heltal $m \leq n$ sådana att det finns två positiva heltal x och y , sådana att $x^2 + y^3 = m$ (dvs, m kan skrivas som en kvadrat plus en kub, t.ex. är 9 ett sådant tal då det kan skrivas som $1^2 + 2^3$).

Exempel: för $n = 10$ är ska funktionen returnera `[2, 5, 9, 10]`.

Krav:

- Listomfattning(ar) behöver vara en central del av din lösning.
- Den returnerade listan behöver vara sorterad och fri från dubletter.
- Du får inte använda rekursion.

Du får använda hjälpfunktioner. Du får använda `filter`, `sort` och `nub`.

(5p)

4. Studera följande kod: `filter (\x-> x `mod` 2 == 1) [1..10]`.

- (a) Vad är typsignaturen för första parametern till `filter`? (2p)
 - (b) Syntaxen som används i den första parametern är döpt efter ett formellt system som presenterades av Alonzo Church på 1930-talet. Namnge syntaxen eller det formella systemet på svenska eller engelska. (1p)
 - (c) Vad blir resultatet av att köra koden i Prelude? (1p)
-

Del 2: Logikprogrammering

5. Kontrollflöde

Betrakta följande Prologpredikat, som vänder på en lista.

```
reverse([], []).
reverse([X | Y], Z) :-
    reverse(Y, Yrev),
    append(Yrev, [X], Z).
```

Predikatet `append(?X, ?Y, ?Z)` är det inbyggda listhanteringspredikatet som är sant om och endast om listan `Z` är lika med konkateringen av listan `X` med listan `Y`.

I din lösning på följande uppgifter ska du behandla `append` som en svart låda och ska inte beskriva kontrollflödet inuti `append`.

- (a) Beskriv kontrollflöde (regelinstanser, unifieringar) steg för steg vid evalueringen av de två första svaren till frågan `reverse(L1, L2)`, samt vad dessa två svar blir. (6p)
- (b) Beskriv kontrollflöde (regelinstanser, unifieringar) steg för steg vid evalueringen av det första svaret till frågan `reverse(X, [a, b])`, samt vad detta svar blir. Du kan använda dig av svaret på (a). (4p)

6. Listor

I denna uppgift ska vi beräkna valresultat. Vi har en lista över atomer som representerar godkända kandidater, t.ex. `[adam, eva]`. Vi har också en lista över röster, där varje röst representeras som kandidaten (en atom alltså) som rösten har gått till, t.ex. `[adam, eva, nisse, eva]`. Notera att det kan finnas röster på icke-godkända kandidater.

- (a) Skriv ett predikat `candVotes(+C, +Vs, -N)` som ska vara sant om och endast om `N` är antalet röster i listan `Vs` som har gått till kandidaten `C`. (5p)

T.ex., frågan:

```
candVotes(eva, [eva, adam, adam, eva, eva, eva, troll], N).
```

ska ge svaret:

```
N = 4.
```

- (b) Skriv ett predikat `electResult(+Cs, +Vs, -Rs)` som ska vara sant om och endast om `Rs` är en lista som sammanfattar antalet röster i listan `Vs` för varje kandidat i `Cs`. Resultatet för en viss kandidat ska representeras som en term `res(C, N)`, där `C` är kandidaten och `N` är antalet röster hen fick. Resultatlistan ska innehålla exakt en sådan term för varje kandidat, i samma ordning som kandidaterna ges i `Cs`. (5p)

Implementationen ska använda predikatet `candVotes(+C, +Vs, -N)` från (a)-uppgiften som en svart låda.

(Se nästa sida för ett exempel.)

T.ex., frågan:

```
electResult([adam, eva, hilma], [adam, eva, nisse, eva], X).
```

ska ge svaret:

```
X = [res(adam, 1), res(eva, 2), res(hilma, 0)].
```

Del 3: Formella språk och syntaxanalys

7. För vart och ett av följande språk, **ange tydligt** om det är reguljärt eller inte (1p per uppgift), och bevisa sedan detta (2p per uppgift).

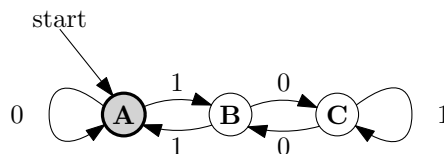
För full poäng på de språk som inte är reguljära krävs ett formellt bevis, men du får använda dig av resultat om reguljära språk som ingått i kursen (t.ex. olika standard-exempel på språk som inte är reguljära).

- (a) Språket med alla strängar över 'a'-'z' som **inte** är palindrom. (3p)
- (b) Språket med alla strängar över '(' och ')' som är balanserade parentesuttryck **och** där längden på strängen är ett primtal. (3p)
- (c) Språket med alla strängar över 0-9 som är ett heltal (i bas 10) som är delbart med 42. (3p)

8. Betrakta följande grammatik över binära strängar, skriven i BNF-notation:

```
<S> ::= "a" <S> "b" <S> | "b" <S> "a" <S> | ""
```

- (a) Ge två exempel på strängar i språket, båda av längd minst 8. (2p)
 - (b) Beskriv med ord (på svenska eller engelska), så lättfattligt och koncist som möjligt, vilket språk som definieras av grammatiken. (2p)
 - (c) Visa att grammatiken är tvetydig, genom att ge en sträng i språket och två olika korrekta syntaxträd för strängen. (3p)
9. Konstruera en grammatik för språket som känns igen av följande ändliga automat (DFA). (4p)



Starttillstånd är **A**, och detta är även det enda accepterande tillståndet.
