Gruppetime 5

Tillukninger og induktivt definerte mengder og rekursive funksjoner

Plan for timen

1. time
- Gjennomgang av denne ukas pensum med eksempeloppgaver

2. time
- Selvstendig jobbing

Nå er innleveringsoppgavene kortet ned

Tillukninger

- Tillukningen av en mengde M
  - Den minste mengden som inneholder M, og er lukket under gitte operasjoner
- Lar A = {1, 2, 3}, R = {<1,2>, <2,1>, <3,1>}
  - Refleksiv tillukning
    - {<1,2>, <2,1>, <3,1>, <1,1>, <2,2>, <3,3>}
  - Symmetrisk tillukning
    - {<1,2>, <2,1>, <3,1>, <1,3>}
  - Transitiv tillukning
    - {<1,2>, <2,1>, <3,1>, <1,1>, <2,2>, <3,2>}
  - **Hva med den transitive og symmetriske tillukningen?**
    - {<1,2>, <2,1>, <3,1>, <1,1>, <2,2>, <3,2>, <1,3>, <2,3>, <3,3>}

Induktivt definerte mengder

- Den minste mengden som inneholder basismengden, og er lukket under gitte operasjoner

- Defineres i 3 steg:
  - Basissteget
  - Induksjonssteget
  - Tillukningen

- Eksempel 1
  - Basismengde: {42, 100} ∈ M. Induksjonssteg: Hvis $x \in M$, så $x-3 \in M$ og $-(2x) \in M$
  - M = {42, 100, 39, -84, 97, -200, ... }

- Eksempel 2
  - M = {2, 4, 16, 256, 65 536, ... }
  - Kan observere at: $2^2 = 4$, $4^2 = 16$ ...
  - Basismengde: {2}. Induksjonssteg: Hvis $x \in M$, så $x^2 \in M$

Lister

Definisjon (Liste)

Mengden av **lister** (*eng: lists*) over en mengde A er induktivt definert som den minste mengden slik at følgende holder:

- $()$ er en liste over A.
  Dette er den **tomme listen** (*eng: empty list*) over A.
- Hvis $x \in A$ og L er en liste over A, er $x :: L$ en liste over A

:: er en funksjon som tar inn et element, x og en liste, L, og putter elementet x foran de andre elementene i lista L.

x **::** L kan leses som x puttet inn i L.

Strenger og konkatenering

Definisjon (Alfabet, tegn, streng)

- Anta at A er en mengde av tegn.
- Mengden A kalles et **alfabet** (*eng: alphabet*).
- Mengden av **strenger** (*eng: strings*) over A er den minste mengden A^* slik at:
  - $\Lambda \in A^*$, hvor symbolet Λ står for **den tomme strengen** (*eng: empty string*) og
  - hvis $s \in A^*$ og $x \in A$, er $sx \in A^*$, hvor sx står for resultatet av å slå sammen s og x.
- Vi skriver s i stedet for Λs.
- Et **språk** (*eng: language*) over A er en delmengde av A^*.

- A = mengden med tegn
- A* = alle mulige strenger over A
- Et språk over A er et utvalg av A*, bestemt av operasjonene til språket

Bitstrenger

Definisjon (Bitstreng)

- Mengden av **bitstrenger** (*eng: bit strings*) er induktivt definert som den minste mengden som er slik at
  - `0` og `1` er bitstrenger.
  - Hvis `b` er en bitstreng, er `b0` og `b1` også bitstrenger.

- Legg merke til at en bitstreng ikke kan være tom.

Kapittel 10 - Rekursive funksjoner

Definisjon (Rekursiv funksjon)

Hvis en mengde M er induktivt definert, kan vi definere en **rekursiv funksjon** (*eng: recursive function*) f med definisjonsområdet M på følgende måte:

- For hvert element x i basismengden til M, spesifiser en verdi for $f(x)$.
- Dette kalles **basissteget** eller **basistilfellet** (*eng: base case*) for funksjonen.
- For hvert element x i M som fremkommer i et induksjonssteg, definer verdien til $f(x)$ ved å bruke de tidligere definerte verdiene for f.
- Dette kalles **rekursjonssteget** (*eng: recursion step*).

Eksempel

- Gitt funksjonen: f(n) = 2n, der n ∈ ℕ
- Siden f er en funksjon på de naturlige tallene, må vi se på hvordan disse er induktivt definert
  - Basissteg: 0 ∈ ℕ
  - Induksjonssteg: Hvis n ∈ ℕ, så er n+1 ∈ ℕ
- Vi må nå definere en rekursiv funksjon basert på basismengden og induksjonssteget til denne mengden
  - Basissteg: f(0) = 0
  - Rekursjonssteget: f(n+1) = f(n) + 2

Tips for dere som foretrekker å regne dere frem til svar fremfor å se etter mønster:

Vi har funksjonen:
f(n) = 2n

Sett inn n+1 for n:
f(n+1) = 2 (n+1)

Regn ut ved hjelp av regneregler:
f(n+1) = 2n + 2

Bytt ut 2n med f(n):
f(n+1) = f(n) + 2

Oppgave 10.3

(a) Definer en funksjon **LEN** på bitstrenger som gir *lengden* på bitstrengen, det vil si antall tegn i den. For eksempel vil **LEN**(0101) = 4.

- Vi må først finne ut hvordan mengden av bitstrenger er induktivt definert.
- Basismengden er {0, 1}, så vi må definere **LEN**(0) og **LEN**(1) som basissteget.
- **LEN(0)= 1, LEN(1)= 1**
- Videre sier definisjonen av bitstrenger at dersom b er en bitstreng, er b0 og b1 bitstrenger. Da må rekursjonssteget definere **LEN**(b0) og **LEN**(b1)
- **LEN(b0)= LEN(b)+1, LEN(b1)= LEN(b)+1**

(b) Definer en rekursiv funksjon **FJERN** på bitstrenger som fjerner alle 0-er i strengen. For eksempel vil **FJERN**(01001010)= 111.

- I denne oppgaven tillater vi å definere funksjonen på mengden av bitstrenger pluss den tomme strengen, for at funksjonen skal være veldefinert på alle bitstrenger.
- Basistilfellet vil da være for den tomme strengen, Λ.
- **FJERN(Λ)= Λ**
- Rekursjonssteget må definere **FJERN**(b0) og **FJERN**(b1)
- **FJERN(b0)= FJERN(b), FJERN(b1)= FJERN(b)1**

Definisjon (Utsagnslogisk formel)

Mengden av **utsagnslogiske formler** (*eng: propositional formulas*) er den minste mengden X slik at følgende holder:

- Enhver utsagnsvariabel er med i X.
  Disse utgjør basismengden og kalles **atomære formler** (*eng: atomic formulas*).
- Hvis F er med i X, er $\neg F$ med i X.
- Hvis F og G er med i X, så er $(F \wedge G)$, $(F \vee G)$ og $(F \rightarrow G)$ med i X.

10.7 Definer en rekursiv funksjon s fra mengden av utsagnslogiske formler til mengden av naturlige tall som er slik at hvis F er en utsagnslogisk formel, er $s(F)$ lik *antall symboler* i F. Hint: Du må være nøye med å ta med alle parenteser. Her er noen eksempler: $s(P) = 1$, $s(\neg P) = 2$, $s((P \to Q)) = 5$, og $s(((P \to Q) \wedge \neg R)) = 10$.

- Det første vi gjør er å finne ut hvordan mengden av utsagnslogiske formler er induktivt definert
  - Basismengden er alle atomære formler P
  - Induksjonssteget er at hvis F og G er formler, så er ¬F, (F ∧ G), (F → G), (F ∨ G) også formler.
- Basissteget:
  - **s(P) = 1**
- Rekursjonssteget:
  - **s(¬F) = 1 + s(F)**
  - **s((F ∧ G)) = 3 + s(F) + s(G)**
  - **s((F → G)) = 3 + s(F) + s(G)**
  - **s((F ∨ G)) = 3 + s(F) + s(G)**