Diskreetti matematiikka, syksy 2010
Matlab-harjoitus 3 (18.11. klo 16-18 MP103)

Tehtäviin vastataan tälle paperille, osoitettuihin tyhjiin alueisiin, yleensä tyhjille riveille. Tehtävät saa – ja on suorastaan toivottavaa – ratkaista parityöskentelynä.

Lataa Kurssimateriaalisivulta `Diskreet2.zip` ja pura se virtuaalilevysi juurikansioon `X:`. Käynnistä Matlab, siirry em. kansioon `X:\DISKREET` sekä aja M-skriptit `alustus` ja `rel`. Nyt sinun pitäisi olla hakemistossa `X:\DISKREET\RELAATIO`. Kokeile `dir`; silloin pitäisi listautua mm. Matlab-demojen 2 tehtävien ratkaisufunktiot `ReflKo.m`, `SymmKo.m`, `AntisKo.m`, `TransKo.m`, `BT.m` ja `EkvivKo.m`. Voit käyttää niitä jatkossa kutsumalla esimerkiksi

```
» M = SatRel(6,6,40); M = M | M';
» EkvivKo(M, '')
» type EkvivKo
```

Kokonaislukuaritmetiikka vs. logiikkaluupit

1. Suorita seuraavat käskyrivit (kun olet yhden kirjoittanut, toiset voi helposti editoida; ylös- ja alasnuolilla voi rullata vanhoja käskyjä):

```
» n=50; M = double(SatRel(n,n)); N = double(SatRel(n,n)); tic,sign(M*N);toc
» n=50; M = SatRel(n,n); N = SatRel(n,n); tic, BT(M,N); toc
» n=100; M = double(SatRel(n,n)); N = double(SatRel(n,n)); tic,sign(M*N);toc
» n=100; M = SatRel(n,n); N = SatRel(n,n); tic, BT(M,N); toc
» n=500; M = double(SatRel(n,n)); N = double(SatRel(n,n)); tic,sign(M*N);toc
» n=500; M = SatRel(n,n); N = SatRel(n,n); tic, BT(M,N); toc
```

   Mitä tässä testattiin ja miten kävi? ____________________

   ____

Relaatioiden sulkeumista

2. Kirjoita Matlab-funktiot

   a) `ReflSulk` ja

   b) `SymmSulk`,

   jotka laskevat ja palauttavat matriisina annetun relaation refleksiivisen ja symmetrisen sulkeuman. Aloita valitsemalla Matlabissa **File**, **New** ja **M-file** sekä kirjoita esimerkiksi (täydennä koodisi alle)

```
function RS = ReflSulk(M);
RS =

function SS = SymmSulk(M);
SS =
```

   Tallenna tuotoksesi funktiotiedostoiksi asianomaisilla nimillä (tarkenteena `*.m`) ja jätä editori käyntiin.

   Aja muutama testi niin, että arvot funktiolla `SatRel` matriisin, lasket funktiollasi sen sulkeumat ja katsot tuloksista sekä tarkastusfunktioilla ovatko oikein. Voit helposti tehdä useita testejä toistamalla vaikkapa käskyrivejä:

```
» M = SatRel(5,5,20), R = ReflSulk(M), S = SymmSulk(M)
» ReflKo(R,''), SymmKo(S,'')
```

3. Tehdään funktio `TrSulk2`, joka laskee ja palauttaa matriisina annetun relaation transitiivisen sulkeuman. Aloitetaan (lyhyyden vuoksi, älä nyt kirjoita kommentteja):

```
function TS = TrSulk2(M);
[m,n] = size(M);
if (m == n)
    M = double(M);      % matriisi numeeriseksi
    I = eye(n);         % yksikkömatriisi
    TS = M;             % alustus, suoraan M
    for i = 1:(n-1)     % periaatteessa TS = M + M^2 + M^3 + .. + M^n,
    TS = sign(M * (I + TS));  % mutta lasketaan vähän nopeammalla tavalla
    end;                 % TS = M(I+M(I+M(...M(I+M)))
    TS = logical(TS);      % positiiviset tosiksi
else disp('relaation matriisin pitää tässä olla nxn')
end;
```

Lopuksi tuote nimelle `TrSulk2.m`.

Testaa nyt funktiota ja sen toiminnan nopeutta erikokoisilla matriiseilla: $n = 10$, $n = 50$, $n = 100$, $n = 150$, $n = 200$, $n = 250$, $n = 300$, ehkä jopa $n = 500$. Voit esimerkiksi toistaa riviä eri arvoilla n:

```
» n = 10; M = SatRel(n,n,10); tic; T = TrSulk2(M); toc, TransKo(T)
```

Testin ja seuraavan tehtävän testin tulokset taulukkoon:

n	10	50	100	150	200	250	300	500
`TrSulk2` aika								
`TrSulk` aika								

4. Vielä nopeammin transitiivinen sulkeuma löytyy koodilla

```
[m,n] = size(M);
TRS = logical(M);
for j = 1:n            % käydään matriisin luvut läpi sarakkeittain
  for i = 1:n          % ylhäältä alas, vasemmalta oikealle
    if  (TRS(i,j) == 1) % jos nuoli  i --> j, yhdistetään i nuolella
      TRS(i,:) = TRS(i,:) | TRS(j,:); % kaikkiin niihin, joihin j:stä on nuoli
    end;
  end;
end;
```

Kirjoita tämä funktioksi `TrSulk.m` ja testaa kuten yllä, tulokset taulukkoon.

5. Kirjoita edellisten avulla Matlab-funktio `EkvSulk`, joka laskee äärellisen relaation matriisista lähtien kyseisen relaation ekvivalenssisulkeuman.

6. Generoi funktiolla `SatRel` sellainen (tai useitakin) 10×10-relaatio $R \subseteq [10] \times [10]$, jonka ekvivalenssisulkeumassa $\overline{R}^e$ on *enemmän kuin kymmenen nuolta, mutta kyseessä ei ole koko joukko* $[10] \times [10]$. Saatat joutua arpomaan useita ehdokkaita, ja valita ykkösten osuuden sopivaksi.

   Piirrä relaatio R ja sen ekvivalenssisulkeuma $\overline{R}^e$ piirtofunktiolla `PiirRel1`. Voit piirtää ensin sulkeuman ja sitten päälle alkuperäisen toisella värillä, esimerkiksi seuraavasti:

» `M=SatRel(10,10,5);T=EkvSulk(M);EkvivKo(T,''),PiirRel1(T),PiirRel1(M,'-g')`

Mitä havaitset nuolikaavioesityksistä, paljonko on hyvä ykkösprosentti? ________________

____

Järjestysrelaatioista

Matlabin funktiota voidaan kutsua vähemmälläkin määrällä parametreja kuin funktion määrittely edellyttäisi. Tämä antaa mahdollisuuden määritellä funktio niin, että sen toiminta riippuu parametrien lukumäärästä. Esimerkiksi funktio `SatRel` toimii jo kahdellakin parametrilla m ja n (rivejä ja sarakkeita), jolloin ykkösten osuudelle käytetään oletusarvoa 50 (prosenttia), mutta mahdollisesti annettu optionaalinen kolmas parametri tulkitaan prosenttimääräksi.
Samoin yllä `EkvivKo`-funktiolle voidaan antaa optionaalinen parametri, vaikkapa tyhje `' '`.
Kutsun sisältämien parametrien määrän ilmoittaa (funktion sisällä) Matlabin muuttuja `nargin` (*number of input arguments*?). Itse funktion toiminta on siis suunniteltava muuttujan `nargin` arvosta riippuvaksi, esimerkiksi käyttäen rakennetta `if (nargin == 3)` ...

---

7. Muodosta seuraavanlaiset Matlab-funktiot, jotka optionaalisella toisella parametrilla kutsuttaessa myös tulostavat tekstinä kyseisen ominaisuuden, jos se on voimassa. Voit tietenkin käyttää jo olemassaolevia funktioitamme.

   a) `JarjKo`, joka lähtötietonaan relaation matriisi palauttaa arvon 1 (tosi), jos ja vain jos relaatio on osittainen järjestys.

   b) `TaysiKo`, joka lähtötietonaan relaation matriisi palauttaa arvon 1 (tosi) jos ja vain jos relaatio on täysi.

   Vihjeeksi: kokeile

```
» M = true(5,5)
» all(all(M | M'))
» M(1,2) = false; M(2,1) = false
» all(all(M | M'))
```

   Testaa funktiotasi! Voit ladata testimatriiseja käskyllä (`load jarjtest.mat`). Millaisia ovat testimatriisien edustamat relaatiot? ______________________________

   ____

8. Muodosta edellisten kanssa yhteensopiva funktio `TaydJarjKo`, joka selvittää onko matriisilla esitettävä relaatio täydellinen järjestys.

   Millaisia ovat testimatriisien edustamat relaatiot? ______________________________

   ____

9. Muodosta funktio `Minimaaliset`, joka etsii järjestysrelaation minimaaliset alkiot, ts. ne, joita pienempiä ei ole. Kokeile koodia:

```
» M = SatRel(10,10)
» M = M & ~diag(diag(M));
» find(~any(M ))
```

   Lataa `relaatio.mat` ja määritä relaatioiden `ojarj` ja `tjarj` minimaaliset alkiot. __________

10. Muodosta funktio `Maksimaaliset`, joka palauttaa järjestysrelaation maksimaaliset alkiot, kun lähtötietona annetaan relaation matriisi. ______________________________

Satunnaisrelaatioiden manipulointia (bonustehtäviä)

10. Muodosta Matlab-funktio `AntisymmSoi`, joka muuttaa (vaikkapa osin satunnaisestikin) annetun $n \times n$-logiikkamatriisin antisymmetriseksi, ts. matriisiksi, joka esittää antisymmetristä relaatiota.

    Voit käyttää koodia:

    ```
    AS = logical(M);
        for i = 1:n             % käydään läpi diagonaalin
            for j = (i+1):n     % yläpuoliset alkiot:
                if (AS(i,j) & AS(j,i))  % jos symmetrisesti ykkösiä, toinen
                AS(i,j) = (rand > 0.5); % nollaksi arpomalla
                AS(j,i) = ~AS(i,j);
    ...
    ```

    Testaa funktiotasi funktiolla `AntisKo`.

11. Muodosta Matlab-funktio `SatAntis`, joka – lähtötietonaan relaation matriisin dimensio ja prosenttiluku (optionaalisena, oletusarvoksi esim. 50), muodostaa "satunnaisen" antisymmetrisen relaation matriisin. Voit käyttää apuna Tehtävän 10 funktiotasi sekä aiempaa funktiota `SatRel`.