

JY / Matematiikan ja tilastotieteen laitos
MATA2500 Informaatioteoria, Syksy 2019
Harjoitus 3

Tämän viikon demoissa puhutaan Kolmogorov-kompleksisuudesta.

1. Muodosta haluamallasi tavalla (Python3:lla tai “järkevällä kuvailulla”) jokin merkkijono jolla on matala Kolmogorov-kompleksisuus, mutta josta kuvailua on mielestäsi hankala löytää.

Selventävä esimerkki:

(a) Numero 9957736225994138912497217752834791315155748572424541506959508295331168617278558890750983817546374649393192550604009277016711390098488240128583616035637076601047101819429555961989467678374494482553797 on kohtuu monimutkaisen näköinen, mutta kyseessä on piin desimaalit 1063-1262.

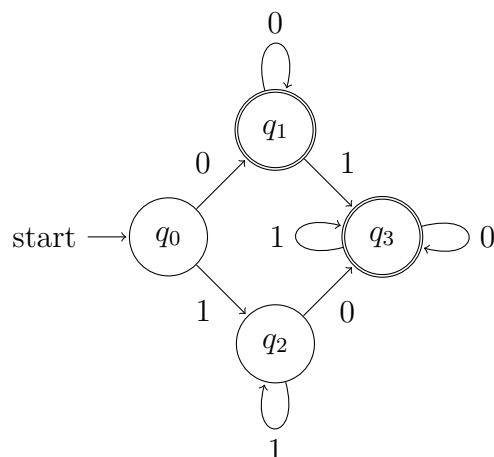
(b) Numero 607913510444246713153276832201611497141106144107457434120999240122502618653111949125517613454773521668186382211648758487881381059414056281181489623069821365111311347251179842731334110461241109145812638036713370132103143577885

On kohtuu monimutkaisen näköinen, mutta sen tuottaa koodi

```
x = 3
for j in range(100):
    x = (199*x + 59) % 149
    print(x, end=" ")
print('')
```

(Taustalla n.k. *Linear congruential generator*.)

2. Määritellään äärellinen automaatti oheisen kuvan mukaan.



Määrittele automaatti formaalisti muodossa $(Q, \Sigma, \delta, q_0, F)$.

Mitkä seuraavista syötteistä ovat sallittuja tälle automaatilelle? Mitä sallitut syötteet tuottavat?

(a) 11000011010110011

(b) 000000000000000000

(c) 000000000000000001

3. Laajenna seuraavia L-systeemejä 3:n askeleen verran.

Systeemi A.

Muuttujat: 0,1

Vakiot: [,]

Aksiooma: 0

Säännöt: $(1 \rightarrow 11), (0 \rightarrow 1[0]0)$

Systeemi B.

Muuttujat: F, X

Vakiot: [,], +, -

Aksiooma: FX

Säännöt: $(X \rightarrow [-FX] + FX)$

Syventävät tehtävät

Seuraavat tehtävät palautetaan kirjallisina harjoitusten pitäjälle, kurssin palautuslokeroon tai sähköpostitse. Deadline 19.11.2019 klo 08.30.

4. Muodosta äärellinen automaatti D jolle löytyy...

- (a) kaksi syötettä a ja b siten, että ab ei ole sallittu syöte.
- (b) kaksi sallittua syötettä x ja y jotka tuottavat tulosteet $p = D(x)$ sekä $q = D(y)$, siten että xy on myös sallittu syöte ja $D(xy) \neq pq$.

5. Lataa tietokoneelle seuraavat tekstitiedostot:

- (a) Iliad ja Odysseia <http://www.gutenberg.org/files/59306/59306-0.txt>
- (b) Moby Dick <http://www.gutenberg.org/files/2701/2701-0.txt>
- (c) Miljoona piin desimaalia https://www.pi2e.ch/blog/wp-content/uploads/2017/03/pi_dec_1m.txt
- (d) Miljoona (pseudo)satunnaista merkkiä <https://luisto.fi/PseudoRandom.txt>

Vertaa kaikissa tiedostoissa alkuperäistä tiedostokokoa sekä kompressoimasi (esim. zip) tiedoston kokoa. Mitä huomaat? Miksi piin desimaalit tiivistyvät lähes yhtä hyvin kuin englanninkielinen teksti?

6. Tämän tehtävän voi suorittaa kahdella eri tavalla.

Quine on tietokoneohjelma, joka tulostaa ulos oman lähdekoodinsa. (Esimerkiksi `print("print(\"hei\\!\")")` ei ole Python-quine, koska se tulostaa `print("hei\\!\")` eikä `print("print(\"hei\\!\")")`.)

Oman quinen rakentaminen on (joidenkin mielestä) todella hauskaa. Tässä tehtävässä idea on rakentaa oma quine valitsemallaan ohjelmointikielellä. (Myös Latex, html tai vastaava formaali ilmaisumuoto käy.) **Mutta!** Quinea ei välttämättä saa aikaiseksi lyhyessä ajassa, ja lisäpisteen himo voi saattaa ihmisen luovuttamaan ja tukeutumaan valmiisiin ratkaisuihin ainoastaan koska aika loppuu kesken. Valmiisiin ratkaisuihin saa toki tukeutua, mutta jos haluaa säästää itselleen aivoja mutkistavan haasteen, niin quinea ei tarvitse saada valmiiksi. Täten tämän tehtävän voi suorittaa kahdella eri tavalla:

- (a) Koodaa Quine, tai
- (b) yritä vähintään tunti tosissasi koodata Quine. Jos et onnistu, niin tunnin jälkeen voit palauttaa tehtävän muodossa “yritin sikana, en saanut tehtyä, mutta haluan yrittää myöhemmin paremmalla ajalla uudelleen ilman spoilereita.”