Kuhunkin kohtaan vastataan siihen TIMissä olevaan laatikkoon, jolla on sama nimi kuin kohdan edessä. **Avustajien käyttö on ankarasti kielletty.** Kurssin aineistoa, muita lähteitä (myös netistä löytyviä), laskimia ja ohjelmia saa käyttää. Laskelmiesi tarkastamiseen kannattaa käyttää sivua `http://users.jyu.fi/%7eava/MathCheck_yleinen.html` Monet tästä PDF-tiedostosta maalaamalla kopioidut kaavat voi pudottaa sinne ja ne kelpaavat syntaksin puolesta sellaisinaan tai pienin muutoksin. Kunkin kohdan maksimipistemäärä on 1. Tentin maksimipistemäärä on 30. Jos tekniikka ei toimi, meilaa `antti.valmari@jyu.fi`

Tarkoittakoon L että logiikka on kivaa, M että matematiikka on kivaa ja O että ohjelmointi on kivaa. Ilmaise seuraavat logiikan merkinnöillä.

L1 Ohjelmointi on kivaa, mutta matematiikka ja logiikka eivät ole.
$O \land \neg M \land \neg L$

L2 Jos matematiikka tai ohjelmointi on kivaa, niin myös logiikka on kivaa.
$(M \lor O) \rightarrow L$

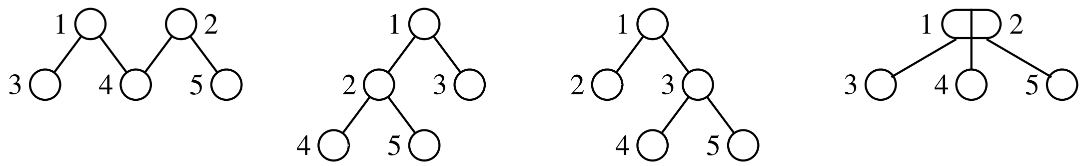
Sievennä seuraavat kaavat mahdollisimman yksinkertaiseen muotoon.

L3 $P \lor \neg Q$
$Q \rightarrow P$

L4 $\neg(-2 \leq x < 6) \land x > 0$
$x \geq 6$

L5 $y \leq 2 \rightarrow x+3 < y \lor x \geq 3y-8$
T

C++:ssa 0 vastaa totuusarvoa false, 1 vastaa totuusarvoa true, `&&` on looginen ”ja” ja `||` on looginen ”tai”. Operaattori `|` ei ole sama kuin `||`, mutta silti se tuottaa luvuilla 0 ja 1 saman tuloksen kuin `||`. (Se on ”bitittäinen tai”, eli se tekee tai-operaation kullekin bittipaikalle erikseen.)



A B C D

P1 Lauseke `0 && 0 || 1` tuottaa 1. Esitä sen lausekepuu valitsemalla jokin kuvista A, B, C tai D ja kertomalla mitä mihinkin solmuun tulee tyyliin A 1:`0` 2:`&&` 3:`0` 4:`||` 5:`1` .
B 1:`||` 2:`&&` 3:`1` 4:`0` 5:`0`

P2 Lauseke `0 && 0 | 1` tuottaa 0. Esitä sen lausekepuu.
C 1:`&&` 2:`0` 3:`|` 4:`0` 5:`1`

P3 Valitse kuvista A, B, C ja D jokin, jota et käyttänyt edellisissä kohdissa. Suunnittele tavallista matematiikkaa käyttävä lauseke, yhtälö, väittämä tms., jonka lausekepuun voi esittää valitsemallasi kuvalla. Kerro vastauksessasi sekä lauseke tms., valitsemasi kuva, että mitä mihinkin solmuun tulee.
$0 \leq i < n$ D 1:$\leq$ 2:$<$ 3:0 4:i 5:n

Y1 Ratkaise epäyhtälö $|4x+y+7| + |2x+2y-4| \leq 0$. Näytä niin paljon välivaiheita tai selosta ratkaisuasi muulla tavoin niin paljon, että näkyy, että osaat ratkaista tämän ilman ohjelmia ja laskimia. Kaksi välivaihetta riittää, jos valitset ratkaisutavan ja vaiheet hyvin.
$|4x+y+7| + |2x+2y-4| \leq 0 \Leftrightarrow 4x+y+7 = 0 \land 2x+2y-4 = 0$
$\Leftrightarrow 4x+4y-8 = 0 \land 3y-15 = 0 \Leftrightarrow x = -3 \land y = 5$

Y2 Ruusa suunnittelee tietoliikenneyhteyttä avaruusalukseen. Vaihtoehdossa A viestitys maksaa 25 € / megatavu ja vastaanotto-ohjelmistoa varten pitää ostaa aluksen tietokoneesta tilaa 1200 €:lla. Vaihtoehdossa B viestitys maksaa vain 18 € / megatavu, mutta avaruusaluksessa pitää olla monimutkaisempi ohjelmisto, jonka tarvitsema tila maksaa 6765 €. Esitä kummankin vaihtoehdon hinta viestitettävien megatavujen määrän n funktiona. Kerro, millä n:n arvoilla B on halvempi.
$25n + 1200$
$18n + 6765$
$18n + 6765 < 25n + 1200 \Leftrightarrow 7n > 5565 \Leftrightarrow n > 795$

B1 Tarkastellaan kielioppia $K ::= \varepsilon \mid aKc \mid bKc$. Anna esimerkki kieleen K kuuluvasta merkkijonosta jonka pituus on 6, tai perustele, että sellaista merkkijonoa ei ole olemassa. Tee sama pituuksille 5, 4, 3, 2, 1 ja 0.
Paritonta pituutta ei voi saavuttaa, koska ε tuottaa nolla ja aKc ja bKc tuottavat kaksi merkkiä. Parilliset: *aaaccc aacc ac* ε

B2 Anna esimerkki kieleen K kuuluvasta merkkijonosta, jossa on vähemmän merkkejä a kuin merkkejä c, tai perustele, että sellaista merkkijonoa ei ole olemassa.
bc

B3 Anna esimerkki kieleen K kuuluvasta merkkijonosta, jossa on enemmän merkkejä a kuin merkkejä c, tai perustele, että sellaista merkkijonoa ei ole olemassa.
Ei ole olemassa, koska aKc on ainoa keino tuottaa a, ja se tuottaa myös c:n.

B4 Aakkosto on $\{a,b,c\}$. Kirjoita BNF-määritelmä kielelle X, joka sisältää ne ja vain ne merkkijonot, joiden ensimmäinen ja viimeinen merkki ovat a, ja mikään muu merkki ei ole a. Lyhimmän tähän kieleen kuuluvan merkkijonon pituus on 1.
$X ::= a \mid aYa$
$Y ::= \varepsilon \mid bY \mid cY$

B5 Kirjoita BNF-määritelmä yksinkertaistetuille C++:n aliohjelmien parametrilistoille. Esimerkkejä parametrilistoista ovat `(int x, char y, int z)` ja `()`. Käytä kielen nimenä P. Tehtävän helpottamiseksi tietotyyppeinä voi esiintyä vain `int` ja `char`, ja parametrien niminä voi esiintyä vain `x`, `y` ja `z`. Sanan `int` perään tulee täsmälleen yksi välilyönti ja samoin sanan `char` ja merkin `,` perään, eikä muualle tule välilyöntejä. Toisin kuin oikeassa C++:ssa, sama nimi saa olla usean parametrin nimi.
P ::= "()" | "(T V)" | "(T VX)"
X ::= "" | ", T VX"
T ::= int | char
V ::= x | y | z

B6 C++:ssa voi antaa parametreille oletusarvoja tyyliin `(int x, char y=74, int z=89)`. Jos jollekin parametrille on annettu oletusarvo, niin myös kaikille sen jälkeen tuleville parametreille on annettava oletusarvo (esim. `(int x, char y=74, int z)` ei ole sallittu). Tehtävän helpottamiseksi oletusarvoina voi käyttää vain `74` ja `89`. Myös kaikki edellisessä kohdassa mainitut helpotukset ovat voimassa. Kirjoita BNF-määritelmä tällaisille C++:n aliohjelmien parametrilistoille. Käytä kielen nimenä Q.
Q ::= "()" | "(X)" | "(Y)"
X ::= "T V"| "T V, X"| "T V, Y"
Y ::= "T V=A"| "T V=A, Y"
T ::= int | char

V ::= x | y | z
A ::= 74 | 89

Taulukko $T[0\ldots n-1]$ indeksoidaan 0:sta $n-1$:een.

T1 Kirjoita kaava, joka sanoo, että T:ssä on ainakin yksi nelonen.
$\exists i; 0 \leq i < n : T[i] = 4$

T2 Kirjoita kaava, joka sanoo, että T on aidosti vähenevässä suuruusjärjestyksessä.
$\forall i; 1 \leq i < n : T[i] < T[i-1]$

T3 Millä i:n arvoilla $0 \leq i \leq n \wedge \forall j; 0 \leq j < i : \exists k; 0 \leq k < n : T[j] < T[k]$ on tosi riippumatta T:n sisällöstä? Perustele vastauksesi.
$i = 0$, koska silloin $\forall$ käy läpi tyhjän välin. Muilla i:n arvoilla joko $0 \leq i \leq n$ ei toteudu tai taulukko, jonka kaikki alkiot ovat samat, tuottaa epätosi.

T4 Olkoon $n > 0$. Millä i:n arvoilla edellisen kohdan kaava on epätosi riippumatta T:n sisällöstä? Perustele vastauksesi.
$i < 0 \vee i \geq n$, koska silloin $0 \leq i \leq n$ on epätosi tai seuraavan kohdan perustelu pätee.

T5 Olkoon $n > 0$. Millä i:n arvoilla $0 \leq i \leq n \rightarrow \forall j; 0 \leq j < i : \exists k; 0 \leq k < n : T[j] < T[k]$ on epätosi riippumatta T:n sisällöstä? Perustele vastauksesi.
$i = n$, koska silloin $\forall$ käy läpi koko taulukon, jolloin taulukon suurimmalle alkiolle $T[j]$ ei voi päteä $T[j] < T[k]$. Muilla i:n arvoilla joko $0 \leq i \leq n$ ei toteudu tai taulukko, jonka viimeinen alkio on muita suurempi, tuottaa tosi.

Seuraavissa kohdissa puhutaan T:n sisällön siirtämisestä vasemmalle siten, että vasemmasta reunasta ulos tulleet alkiot lisätään loppuun. Jos esimerkiksi $T = [1,2,3,4,5]$, niin siirtäminen kaksi askelta vasemmalle tuottaa $[3,4,5,1,2]$.

T6 Anna järkevä tulkinta ilmaukselle ”siirretään -1 askelta vasemmalle”. Mitä siirtäminen -1 askelta vasemmalle tuottaa taulukosta $[1,2,3,4,5]$?
Se tarkoittaa samaa kuin siirretään yksi askel oikealle. Se tuottaa $[5,1,2,3,4]$.

T7 Mitä siirtäminen 654321 askelta vasemmalle tuottaa taulukosta $[6,5,4,3,2,1]$?
$[3,2,1,6,5,4]$

T8 Kirjoita kaava, joka sanoo että T:n sisältö säilyy muuttumattomana, jos se siirretään i askelta vasemmalle. Esimerkiksi kun $i = 3$ ja $T = [1,0,0,1,0,0]$ se tuottaa tosi mutta kun $i = 3$ ja $T = [1,0,0,2,0,0]$ se tuottaa epätosi.
$\forall j; 0 \leq j < n : T[(i+j) \bmod n] = T[j]$

Oheista algoritmia tai sen muunnoksia käytetään muun muassa julkisen avaimen salakirjoituksen avainten luonnissa. Kaikkien muuttujien tyyppi on kokonaisluku. Merkitsemme muuttujien n ja m alkuperäisiä arvoja N ja M. Oletamme, että $N \geq 0$ ja $M \geq 0$. Voidaan todistaa, että aina rivin 2 alussa pätee $n = a_1N + a_2M \wedge m = b_1N + b_2M$.

```
1   a1 := 1;  a2 := 0;  b1 := 0;  b2 := 1;
2   while m ≠ 0 do
3       d := n div m;
4       k := a1 − d · b1;  a1 := b1;  b1 := k;
5       k := a2 − d · b2;  a2 := b2;  b2 := k;
6       k := n mod m;  n := m;  m := k;
```

O1 Mitä muuttujassa m on kun ohjelma lopettaa?
0

O2 Mitä muuttujassa n on kun ohjelma lopettaa?
$\gcd(N,M)$

O3 Miksi $n = a_1N + a_2M \wedge m = b_1N + b_2M$ on voimassa, kun riville 2 tullaan ensimmäisen kerran?
Rivin 1 ansiosta $a_1N + a_2M = 1N + 0M = N \wedge b_1N + b_2M = 0N + 1M = M$. Koska kerta on ensimmäinen, n:n ja m:n arvoja ei ole vielä muutettu, joten symbolien N ja M määritelmien ansiosta $n = N$ ja $m = M$.

O4 Miksi $n = a_1N + a_2M$ on voimassa rivin 6 lopussa?
Muuttujien n, a_1 ja a_2 arvot rivin 6 lopussa ovat samat kuin muuttujien m, b_1 ja b_2 arvot rivin 2 alussa, ja invariantti lupaa että $m = b_1N + b_2M$ on voimassa rivin 2 alussa.

O5 Miksi $m = b_1N + b_2M$ on voimassa rivin 6 lopussa?
Rivin 2 alun muuttujien arvoilla ilmaistuna se sanoo $n \bmod m = (a_1 - (n \operatorname{div} m)b_1)N + (a_2 - (n \operatorname{div} m)b_2)M = a_1N + a_2M - (n \operatorname{div} m)(b_1N + b_2M)$. Se saadaan invariantilla kaavasta $n \bmod m = n - (n \operatorname{div} m)m$, joka pätee jakoyhtälön vuoksi.

Z1 Jos kurssille laaditaan ensi vuodeksi yksi uusi tehtävä, niin mitä aihetta sen pitäisi sinun mielestäsi käsitellä? Miksi juuri sitä aihetta? Pitäisikö uuden tehtävän olla vaihtoehtoinen vai pakollinen? Jos tulee uusi pakollinen tehtävä sinun aiheestasi tai jostain muusta aiheesta, niin, jotta pakollisten määrä ei kasvaisi, mikä vanha pakollinen tulee muuttaa vaihtoehtoiseksi?

loppu