KONSTRUKTSIOONID JA ALGORITMID

EBS

Konstruktsioonid ja algoritmid

- Struktuursete konstruktsioonide **moodustamine** toimub **samm-sammult** rangelt kindlaks määratud valdkonnas ja rangelt **kindlaks määratud korra kohaselt**
- Kindlaks määratud valdkonnas kindlaksmääratud viisil korrastatud sammude (tegevussammude) kogumi ***esitust*** kutsutakse sageli **eeskirjaks** ehk **algoritmiks**

Algoritmiks ehk eeskirjaks olemise korral eeldatavad üldised tunnused

diskreetsus – tegevused, sh üksikud tegevussammud, nende tegevuste objektid (ehk “lähtematerjal” ning tulemused) ja tegevuste sooritamise ajad ehk tegevuste asendid nende tegevuste järjestuses peavad kõik olema **võimalikult selgelt eristatavad** (st et üks tegevus peab olema selgelt eristatav teisest tegevusest, üks objekt peab olema selgelt eristatav teisest objektist ning üks aeg peab olema selgelt eristatav teisest ajast)

elementaarsus – üksikute tegevussammude iseloom ja nende kirjeldus peab olema **võimalikult lihtne**

direktiivsus – iga üksiku sammu korral – ka siis, kui mingil põhjusel pole seda sammu võimalik sooritada – **peab olema selge**, kas tegevused lõpetatakse või kui jätkatakse, siis milline on järgmine tegevussamm. Tavaliselt on antud sammu korral, kui kirjas pole teisiti, järgmisena sooritatavaks sammuks see samm, mille kirjeldus paikneb antud sammu kirjelduse järel.

determineeritus – iga tegevussammu korral peab tulemus olema **võimalikult üheselt määratud** sellega, millest vastava sammu sooritamisel lähtutakse

reprodutseeritavus – sammud ja nende kogumid **peavad olema korratavad**

Algoritmi mitteformaalne määratlemine

Mitteformaalne määratlus.
Diskreetsuse, elementaarsuse, direktiivsuse, determineerituse ja reprodutseeritavuse tingimusi rahuldava **sammude kogumi** ***esitust*** nimetame ***eeskirjaks*** (ehk ***juhendiks*** ehk ***juhtnööriks*** ehk ***instruktsiooniks*** ehk ***algoritmiks*** ehk ***programmiks*** vms). Samm-sammulisi tegevusi, mis toimuvad kooskõlas vastavate eeskirjadega, nimetame ***eeskirjastatud tegevuseks*** (vahel ka ***juhendite ehk instruktsioonide kohaseks tegevuseks*** või näiteks ***algoritmeeritud tegevuseks*** või siis ***programmeeritud tegevuseks***)

Märkus. Tundub, et ülalesitatud ***eeskirja mitteformaalne määratlus on mõistlikul viisil kooskõlas eespool määratletud struktuursete konstruktsioonide ja esitustega*****.**

Algoritm ja al-Horazmi

Terminid ***algoritm*** ja ***algoritmiteooria*** on seotud Horezmist pärit ja Bagdadis töötanud kuulsa araabia õpetlase **al-Horazmi**, täpsemalt – **Muhhammad ibn Musa al-Khwarizmi** (u. 783 – – u. 850) nimega. Arvatakse, et tänu just tema tööde ladinakeelsetele tõlgetele (näiteks "***Algorizmi de numero Indozum***") levisid Lääne-Euroopasse Indiast pärit positsiooniline kümnendsüsteem koos vastava numeratsiooni (nn *araabia numbrid*) ning arvutuseeskirjadega.

al-Horazmi matemaatika alastes teostes olid poeetiliselt sõnastatud näidete vormis esitatud muuhulgas ka juhised kuut tüüpi ruutvõrrandite ($ax = b$, $x^2 = a$, $x^2 = ax$, $x^2 = ax + b$, $x^2 + a = bx$, $x^2 + ax = b$) lahendamiseks. Erinevate autorite teostes hakkas, nagu tol ajal ikka ette tuli, araabiakeelne nimi tasapisi teisenema. Seetõttu kogusid lisaks **Algorizmile** kuulsust ka **Algorismus** ja **Algorithmus**

Algoritmeerituse võlud ja valud (I)

Eeskirjade koostamine ja nende sihiteadlik täitmine on kujunenud ning arenenud inimkooslustes aastatuhandete vältel. Tehnikas on see väldanud ehk mõnevõrra vähem, kuid see-eest on jõudnud niivõrd imetlusväärsete tulemusteni, mis inimsoo teadvusesse tagasi peegeldudes on ühest küljest *olulisel määral korrastanud ja selgitanud* ***eeskirjade olemust*** ning eeskirjastatud tegutsemise võlusid, teisest küljest aga tekitanud teatava usu, et

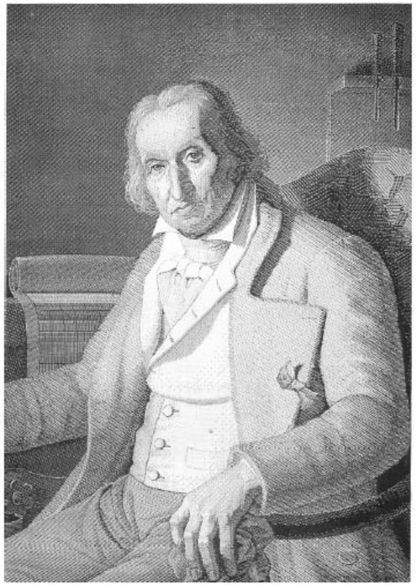
(A) kõik see, mis *mõistlikult*, hästi ja tulemuslikult kulgeb, on korralikult eeskirjastatud tegevuse vili

(B) kui miski on *ebamõistlik* või midagi untsu läheb, siis on reeglina tegemist eeskirjade eiramise või kuidagi kehvalt koostatud eeskirjadega (seda viimast muidugi haruharva, kuna eeskirjad on üldiselt nii jumalikud asjad, mille ebatäiuslikkusest tavakodanikud mõeldagi ei tohiks)

Tõele au andes on äsjanimetatud usul (*mõistlikkus* ⇔ ⇔ *korralik eeskirjastatus*) võrdlemisi tugevad alussambad.

Algoritmeerituse võlud ja valud (II)

<u>**Näited tehnika ajaloost**</u>


Joseph Marie Jacquard

- mitmesugused tekstiilitööstuse masinad (sh aastal 1805 prantsuse leiduri **Joseph-Marie Jacquardi** (1752 – 1834) poolt konstrueeritud keerulise- ja suuremustrilise kanga kudumiseks mõeldud masin, mille töö juhtimiseks vajalikud programmid olid "kirjutatud" papist kaartidele vastavate augukeste kombinatsioonide abil)
- muusikamasinad (mis esitasid vastaval viisil augustatud metallketastelt "instruktsioone lugedes" tol ajal populaarseid lugusid nö ühest ja samast kastist, milles paiknesid kavalalt konstrueeritud keelpillid, puhkpillid ja isegi väikesed trummid)
- kindlatel aegadel terveid nukuetendusi andvad tornikellad (näiteks Müncheni raekojas) ning palju-palju muud.

Algoritmeerituse võlud ja valud (III)

“Inimesemoodi” toimivad – sh ***eeskirju täpselt täita suutvad***, või vähemalt inimese tegevusi täpselt kordavad, kuid inimestega võrreldes suhteliselt võimekad, mittetõrkuvad ja vähenõudlikud masinad ei piirdunud ainult ebameeldiva alaväärsustunde tekitamisega, vaid ähvardasid inimesi tööst ja leivast ilma jätta. Sellest asjaolust tulenesid vahel üsnagi karmid “vastukäigud”.

Näide. Üks mõtlemapanev lugu on seotud nn ***paelakudumismasinaga***, mis võimaldas kududa samaaegselt mitut paela, kandes kuduja iga liigutuse üle korraga kõigile kudumisel olevatele paeltele. (NB! Paneme siinkohal tähele, et kõnealune masin olevat suutnud ühes kohas jälgida, mida inimene teeb ning seda sama masina teistes piirkondades täpselt korrata.)

Tegemist olevat olnud üsna keerulise masinaga, mis väidetavalt leiutati Gdanskis aastal 1579. **Linnavõimud, kes kartsid kangrute töötajäämist, keelasid avalikkust leiutisest teavitada ning lasksid leiduri salamahti surnuks kägistada**

Algoritmide näited (I)

Johnsoni pesuvahu kasutamise juhend vaiba puhastamiseks.

1. samm: puhasta vaip tolmuimejaga
2. samm: kata kinni mööbli jalad
3. samm: raputa purki korralikult
4. samm: pritsi 10 cm kauguselt ühtlane vahukiht kuni 1 m^2 suurusele pinnale
5. samm: hõõru vaht niiske käsnaga puhastatavasse pinda
6. samm: vajaduse korral loputa vahepeal käsna
7. samm: harja vaip ühesuunaliste liigutustega üle (et karv püsti jääks)
8. samm: lase vaibal kuivada 2 – 4 tundi (veel parem – terve öö)
9. samm: eemalda tolmuimejaga lahtitulnud mustus
10. samm: **kui** puhastatav koht on ikka määrdunud, siis suundu sooritama sammu nr 2.,

    kui kavandatud siht (puhastada määrdunud koht) on saavutatud, siis suundu sooritama sammu nr 11.
11. samm: lõpeta antud tegevus.

Algoritmide näited (II)

Täisarvulise ruutjuure otsimise algoritm etteantud arvust A.

1. samm: võta suuruseks B arv 0.
2. samm: arvuta korrutis $B \cdot B$
3. samm: võrdle arvu A korrutisega $B \cdot B$
4. samm: **kui** arvud A ning $B \cdot B$ on võrdsed, siis suundu täitma sammu 6.

   kui arvud A ning $B \cdot B$ ei ole võrdsed, siis suundu täitma sammu 5.
5. samm: suurenda suurust B ühe võrra ning suundu täitma sammu 2.
6. samm: kuuluta arvu A täisarvuliseks juureks suuruse B arvuline väärtus
7. samm: lõpeta antud tegevus

Märkus. **Pole välistatud**, et üht ja sama eeskirja täites võib juhtuda, et kuigi igat sammu saab tulemuslikult sooritada – **pole eeskirja täitmisel võimalik jõuda … lõpptulemuseni!**

Algoritmide näited (III)

Üksliikmete (ehk arvudest ja muutujatest moodustatud korrutiste) korrastamise eeskiri

(1) Vaheta üksliikmes esinevate arvude ja muutujate asukohad nii, et kõik arvulised tegurid paikneksid enne muutujaist tegureid
(2) Asenda eelnimetatud tegurite kogum ühe uue teguriga, milleks on eespoolnimetatud arvuliste tegurite korrutis
(3) Vaheta muutujaist tegurite asukohad nii, et ükski vaadeldav muutuja ei paikneks kahe sellise muutuja vahel, mis teineteisest ei erine, küll aga erinevad vaadeldavast muutujast
(4) Moodusta samadest muutujatest tegureid omavahel korrutades vastavad astmed
(5) Vaheta eespool nimetatud astmetest kujunenud tegurite asukohad nii, et astmed paikneksid astmenäitajate kahanevas või vähemalt mittekasvavas järjestuses **(6)** Lõpeta tegevus.
Näide. **$uw6uy^2z3y^3p^2z^24$**, $6\cdot3\cdot4uwuy^2zy^3p^2z^2$, $72uwuy^2zy^3p^2z^2$, $72uuwy^2y^3p^2zz^2$, $72u^2wy^5p^2z^3$, **$72y^5z^3u^2p^2w$**

Sümbolid ja sõnad

- Vaatleme hulka A, mida ennast nimetame kokkuleppeliselt **alfabeediks** ja mille elemente nimetame kokkuleppeliselt **sümboliteks**.

  Määratlus 1. Hulga A iga sümbolit ja iga korteeži, mille positsioneeritud elementideks on alfabeedi A elemendid, nimetame **alfabeedi A sõnaks** ehk lihtsamalt – **sõnaks.** "Ühesümbolilise" sõna pikkus on 1. Ülejäänud sõnade pikkuseks on vastava korteeži pikkus.
- **Kokkulepe.** Kui ⟨a,b,…,c⟩ on sõna, siis **ab…c ∫ ⟨a,b,…,c⟩**
- **Määratlus 2.** Kui **ab…c** on sõna ja **xy…z** on sõna, siis sõna **ab…cxy…z** on sõnade **ab…c** ja **xy…z** **konkatenatsioon.**
- **Kokkulepe.** **ab…c©xy…z = ab…cxy…z**
  ⟨a,b,…,c⟩©⟨x,y,…,z⟩ = ⟨a,b,…,c,x,y,…,z⟩

Alfabeedi A sõnade c-struktuur

- **Määratlus.** Olgu L_A alfabeedi A kõikide sõnade hulk. Süsteemi $\langle L_A; © \rangle$ nimetame **alfabeedi A sõnade konkatenatsioonistruktuuriks** ehk lühemalt **c-struktuuriks.**
- **Teoreem.** Kui $\langle L_A; © \rangle$ on alfabeedi A c-struktuur, siis selle baasiks on hulk A.

Osasõnad ja nende esinemised

- **Kokkulepe.** Olgu nii, et eksisteerib ilma sümboliteta sõna ehk **tühisõna**, mis kokkuleppeliselt **esineb** igas sõnas **osasõnana**
  (1) vahetult esimese sümboli ees ja
  (2) vahetult viimase sümboli järel ja
  (3) kõrvuti paiknevate sümbolite vahel, kui sõna pikkus on vähemalt 2.

  Määratlus. Sõna X on sõna W **osasõna**, mis **esineb** sõnas W, kui

  (0) X on tühisõna või

  (1) leidub selline sõna W_1, mille korral $W=XW_1$ või
  (2) leidub selline sõna W_2, mille korral $W=W_2X$ või
  (3) leiduvad sellised sõnad W_3 ja W_4, mille korral $W=W_3XW_4$.
  Kui seejuures (1)&(2) või (1)&(3) või (2)&(3) või (1)&(2)&(3) ja vastavad W_1,W_2,W_3,W_4 pole tühisõnad, siis kõneleme osasõnal X on sõnas W **mitu esinemist** (mis on **vastavalt järjestatud**)

Asendusseosed sõnade vahel

Määratlus. Tähistagu W′,W″,X,Y sõnu. Ütleme, et **sõnad W′ ja W″ on asendusseoses X→Y,** kui
(1) sõnas W′ esineb osasõna X ja
(2) sõnas W″ esineb osasõna Y ja
(3) sõna W″ on saadav sel teel, et osasõna X mingi üks esinemine sõnas M asendatakse sõna Y esinemisega. Seejuures kirjutame (X→Y)(W′,W″)

Märkus. Määratluses nimetatud sõnad võivad osutuda tühisõnadeks. See pole keelatud!

Näited.
(ta→ka)(tava**ta**rgutaja,tava**ka**rgutaja),
(ta→ka)(**ta**vatargutaja,**ka**vatargutaja),
(tava→)(**tava**targutaja, targutaja),
(→d)(tavatargutaja,tavatargutaja**d**).

Thue süsteemid (I)

- Vaatleme alfabeeti A ning selle alfabeedi kõikide sõnade hulka L_A.

Määratlus 1. Süsteemi $\langle L_A ; \Sigma \rangle$ nimetame **Thue süsteemiks** (Norra matemaatiku **Axel Thue** (1863 – 1922) nime järgi), kui selle signatuur Σ koosneb mingitest fikseeritud asendusseostest hulga L_A elementide vahel.

Määratlus 2. Sõna W nimetame **töödeldavaks** süsteemis $\langle L_A ; \Sigma \rangle$, kui signatuuris Σ leidub selline seos $X \rightarrow Y$, mille korral W sisaldab osasõna X. Vastupidisel juhul ütleme, et sõna M **pole töödeldav** süsteemis $\langle L_A ; \Sigma \rangle$.

Thue süsteemid (II)

Määratlus 3. Ütleme, et Thue süsteem $\langle L_A ; \Sigma \rangle$ **töötleb sõna W sõnaks V**, kui
(1) sõna W on selles süsteemis töödeldav ja
(2) sõna V on mittetöödeldav ja
(3) selles süsteemis leidub niisugune konstruktsioon, mille algkomponendiks on W ja tulemuseks on V.

Näide 1. Vaatleme Thue süsteemi $\langle L_{\{x,y,+,–\}} ; +x–y \to –y+x, +x–x \to, +y–y \to \rangle$. Selline süsteem töötleb sõna +y+x+x–y–y–x sõnaks –y+x.

Näide 2. Vaatleme Thue süsteemi $\langle L_{\{A,B,\cup\}} ; A \cup A \to A, B \cup B \to B, A \cup B \to B \cup A\rangle$. Selline süsteem töötleb sõna $A \cup B \cup A \cup B \cup B \cup A$ sõnaks $B \cup A$.

Markovi normaalalgoritmid (I)

- Fikseerime lõpliku sümbolite hulga A, mis **ei sisalda** sümboleid $\rightarrow$ ning $\rightarrow\bullet$
- Vaatleme selliseid lõplikke jadasid ehk ***sõnu***, mis koosnevad ainult hulka A kuuluvatest sümbolitest
- Võtame kasutusele kahte tüüpi seosed sõnade vahel, mida nimetame ***lihtsateks asendusteks*** ning ***lõpetavateks asendusteks*** ja tähistame need vastavalt kirjutistega $\alpha\rightarrow\beta$ ning $\alpha\rightarrow\bullet\beta$, kus α ning β on mingid fikseeritud sõnad

- Asendustest moodustatud lõplikku jada nimetame ***Markovi normaalalgoritmi skeemiks*** vene matemaatiku **Andrei Markovi** (1903 – 1979) auks.

Markovi normaalalgoritmid (II)

- Sõnu **S** ja **N** loeme **seotuks seosega** $\alpha \to \beta$ või $\alpha \to \bullet\beta$, ***kui***

  - sõnas **S** esineb osasõna α

  - sõnas **N** esineb osasõna β

  - sõna **N** on saadav sõnast **S** sel teel, et, osasõna α ***esimene esinemine*** asendatakse osasõnaga β.

- **Niisugusel juhul** ütleme, et vastav **asendus** $\alpha \to \beta$ või $\alpha \to \bullet\beta$ ***on rakendatav*** sõnale **S** ja veel ütleme, et sõna **N** on vastava asenduse ***rakenduse tulemuseks*** (sõnale **S**) või ütleme veidi lihtsamalt, et sõna **N** on saadav sõnast **S** vastava asenduse abil

Näide 1. Rakendades asendust $10 \to 01$ sõnale 110101, saame tulemuseks sõna 101101.

Näide 2. Rakendades asendust $\to \heartsuit\heartsuit$ sõnale $\diamondsuit\spadesuit\spadesuit$ saame sõna $\heartsuit\heartsuit\diamondsuit\spadesuit\spadesuit$, kuna tühisõna ***esimene esinemine*** sõnas $\diamondsuit\spadesuit\spadesuit$ paiknes enne sümbolit $\diamondsuit$.

Markovi normaalalgoritmid (III)

- Skeemiga **K** normaalalgoritmi ***rakendamine*** ehk ***töö sõna S korral*** koosneb järgnevatest ***sammudest*** **:**
  sammu 0 tulemuseks on sõna **S**

  sammu n tulemuseks **olgu** nüüd sõna $\mathbf{S_n}$
  sammu n+1 tulemuseks **on** samuti sõna $\mathbf{S_n}$, ***kui*** skeemis ***ei leidu*** ühtegi asendust, mida saaks rakendada sõnale $\mathbf{S_n}$. ***Niisugusel juhul*** öeldakse, et **algoritm lõpetab oma töö tulemusega $\mathbf{S_n}$.**

  sammu n+1 tulemuseks **on** sõna $\mathbf{S_{n+1}}$, ***kui*** see sõna on saadav sõnast $\mathbf{S_n}$ skeemis **K** sisalduva ***esimese sobiva asenduse*** rakendamise teel. ***Kui seejuures*** osutub, et tegemist on ***lõpetava asendusega***, ***siis*** **algoritm lõpetab oma töö tulemusega $\mathbf{S_{n+1}}$.** ***Kui*** ühegi n+1 korral algoritm **ei lõpeta** oma tööd, ***siis*** öeldakse, et **sõna S korral on tulemus määramatu.**

Normaalalgoritmide näited (I)

- Vaatleme algoritmi, mille skeemis on ainult üks (lõpetav !) asendus:

  $\to\bullet$

  Selle asendatakse korral tühja osasõna esimene esinemine uuesti tühja osasõnaga. Antud algoritm võimaldab arvutada arvutada samasusfunktsiooni (näiteks I_{11}).
- Vaatleme algoritmi, mille skeemis on ainult üks (lihtne !) asendus:

  $\to$

  Antud juhul, kuna tegemist **pole** lõpetava asendusega, töötab niisugune algoritm suvalise sõnaga lõputult ja seetõttu arvutab mitte kusagil määratud funktsiooni.

Normaalalgoritmide näited (II)

- Vaatleme algoritmi, mis töötleb sümbolitest 0,1,2 koosnevaid sõnu ja mille skeem on selline:
  $20 \to 02,\ \ 21 \to 12,\ \ 2 \to \bullet 012,\ \ \to 2.$
  Saab näidata, et antud algoritm töötleb iga sõna S , mis koosneb ainult sümbolitest 0 ja 1 sõnaks S012. Rakendame vaadeldavat algoritmi sõnadele
  001 ning 110:

SAMM	ASEND.	TULEM.	SAMM	ASEND.	TULEM.
0	-	001	0	-	110 .
1	$\to 2$	2001	1	$\to 2$	2110 .
2	$20 \to 02$	0201	2	$21 \to 12$	1210 .
3	$20 \to 02$	0021	3	$21 \to 12$	1120 .
4	$21 \to 12$	0012	4	$20 \to 02$	1102 .
5	$2 \to \bullet 012$	001012	5	$2 \to \bullet 012$	110012.

Markovi tees ehk normaliseeritavuse printsiip

- Kõik, mis on arvutatav mingi normaalalgoritmi abil – on algoritmiliselt arvutatav
- Kõik, mis on algoritmiliselt arvutatav – on arvutatav sobiva normaalalgoritmi abil

Mitte-eeskirjastatavus

- Algoritmiteoorias on tõestatud, et isegi põhikooli aritmeetika tasemel võivad tekkida äärmiselt süütu väljanägemisega ülesanded ja **probleemid**, mille lahendamiseks vajaliku eeskirja leidmine tundub olevat käkitegu ja seda olukorras, kus tegelikult … **vastavat algoritmi põhimõtteliselt polegi olemas.**

- **Määratlus.** Ülesannet või **probleemi** nimetame ***algoritmiliselt mittelahenduvaks***, kui sobivat lahendusalgoritmi **ei eksisteeri**.

- **Märkus1.** Juhime tähelepanu sõnadele ***ei eksisteeri.*** See tähendab, et jutt pole mitte seni veel loomata algoritmist või ülesleidmata eeskirjast, vaid asjast, mida põhimõtteliselt ei eksisteeri.
- **Märkus 2.** Ülesande *algoritmilise lahenduvuse* **tõestamiseks** piisab sellest, kui esitada sobiv lahendusalgoritm. Ülesande *algoritmilise mittelahenduvuse* **tõestamiseks** tuleb tõestada, et antud ülesande lahendamiseks sobivat **algoritmi ei eksisteeri.**

Karakteriseerimise probleem

- **Täieliku karakteriseerimise probleem:** Leida uuritava hulga H korral selline **algoritmiliselt** arvutatav funktsioon ehk **karakteriseeriv funktsioon** k, mille korral
- $k(x) = 1$, kui $x \in H$
- $k(x) = 0$, kui $x \notin H$
- **Osalise karakteriseerimise probleem:** Leida uuritava hulga H korral selline **algoritmiliselt** arvutatav funktsioon ehk **osaliselt karakteriseeriv funktsioon** k, mille korral
- $k(x) = 1$, kui $x \in H$
- $k(x)$ **on määramatu**, kui $x \notin H$

Algoritmiline mittelahenduvus

Näide 1. Formaalse aritmeetika absoluutselt õigete valemite ära tundmiseks puudub algoritm

Näide 2. Pole olemas algoritmi selleks, et “koodi järgi” eristada resultatiivseid programme mitteresultatiivsetest

Genereerimise probleem

- **Leida etteantud hulga H jaoks niisugune algoritmiliselt arvutatav funktsioon g, mille väärtuste piirkonnaks on hulk H. Ehk mille korral** $H = \{ z \mid (\exists m \in N)[z=g(m)] \}$

Teoreem. Suvalise hulga **genereerimise probleem** on **samaväärne** selle hulga **osalise karakteriseerimise probleemiga.**

Ehk teiste sõnadega: kui leidub selline algoritmiliselt arvutatav funktsioon k, mille korral

- $k(x) = 1$, kui $x \in H$
- $k(x)$ **on määramatu**, kui $x \notin H$,

siis leidub ka selline algoritmiliselt arvutatav funktsioon g, mille korral

- $H = \{ z \mid (\exists m \in N)[z=g(m)] \}$
- **Ja vastupidi!**

Algoritmiline mittegenereeritavus

<u>Näide 1.</u> Formaalse aritmeetika kõikide täiesti õigete valemite hulk pole algoritmiliselt genereeritav.

<u>Näide 2.</u> Esimest järku predikaatarvutuse kõikide mitte-täiesti õigete valemite hulk pole algoritmiliselt genereeritav

MITTE-EESKIRJASTATAVUSE PRINTSIIP:

Leidub niisuguseid ülesandeid ja probleeme, mille lahendamiseks põhimõtteliselt puuduvad vastavad algoritmid, programmid, eeskirjad jms.