FORMAALI SYSTEEMI (in Nutshell):

Formaali kieli:

aakkosto: alkeismerkkien joukko

kieliopin määräämä syntaksi: *sallittujen merkkijonojen* rakenne, formaali kuvaus esim. SSM:n tai EBNF:n avulla

Semantiikka: *sallittujen merkkijonojen* tulkinta eli mitä ne “oikeasti” tarkoittavat kyseisessä tapauksessa.

Päättelymekanismi:

aksioomat: *sallittujen merkkijonojen* perusjoukko, “kantatotuudet”

päättelysäännöt: miten aksioomista saa muodostaa uusia “totuuksia” välittöminä seurauksina

Lauseet ja todistukset:

todistus: aksioomista ja olemassaolevista *sallituista merkkijonoista* päättelysääntöjen avulla muodostettu äärellinen ketju

lause: todistuksen seurauksena muodostunut uusi *sallittu merkkijono*

Johto: joukosta premissejä johdettu uusi *sallittu merkkijono*

Formaalit kielet:

Normaalien kielten analysointi *kolmella eri tasolla*:

1. leksikaalisella tasolla sanat liitetään kielelle määritellyn sanaston perustyyppeihin, esimerkiksi ’auto’ substantiivi mutta ’umpsutäh’ luultavasti kirjoitusvirhe

2. syntaktisella tasolla kuvataan sitä, kuinka sanoista voidaan muodostaa sallittuja lauseita

3. semanttisella tasolla muodostettuihin lauseisiin liitetään niiden todellinen merkitys

ohjelmointikielten rakenne periaatteessa hyvin samankaltainen kuin normaaleissa kielissä, paitsi lausekkeiden semanttinen tulkinta tapahtuu aina ohjelmointikielen määräämässä kontekstissa; leksikaalisia elementtejä: tunnisteet, literaalit, erikoissanat (avainsanat), operaattorit, kommentit, erottimet, tyhjät merkit

Aakkosto (alphabet):

- kielen alkeismerkit, esim. Σ = {a, b, ..., z }
- yleisin merkintä isot kreikkalaiset kirjaimet $\Sigma, \Gamma, \ldots$
- yleisin merkintä merkkien määrälle eli aakkoston koolle $\mid \Sigma \mid$ = 25
  myös merkkijonon pituus $\mid '\mathrm{abc}' \mid = 3$
- esimerkkejä:
  - Reaaliluvut, esim. “210.542”, muodostuvat aakkoston {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, .} symboleista.
  - Binääriluvut, esim. “100101”, muodostuvat aakkoston {0, 1} symboleista.
  - Pelikortit, esim. “♣10”, muodostuvat aakkoston {♣, ♢, ♡, ♠, A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K} symboleista.
  - Luentomonisteen eri lukujen ja kappaleiden numerointi, esim. “2.5.3”, muodostuu aakkoston {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, .} symboleista.
- *huomataan:* kahdella eri formaalilla kielellä voi olla sama aakkosto tai toisen aakkosto voi olla toisen aakkoston osajoukko

kielioppi (grammar): minkälaisia merkkijonoja aakkostosta voi muodostaa

- määrittää *syntaksin* eli ne säännöt, joiden mukaan aakkosista saa muodostaa kielen sisältämiä *sallittuja merkkijonoja* (well formed formula = wff)
  $\Rightarrow$ *formaali kieli* = { *sallitut merkkijonot* }

ESIMERKKI KIELIOPISTA:

Kielen $\mathcal{L}$ aakkoston muodostavat symbolit {★, ◇}. Kielen syntaksin määrittää kielioppisääntö:

Kielen $\mathcal{L}$ *sallitut merkkijonot* muodostuvat nollan tai äärellisen mittaisesta jonosta symboleja ★, joita seuraa yhdestä neljään symbolia ◇ tai yhden tai usemman symbolin ★ jonosta, jota ei seuraa symboleja ◇.

Tällöin kielen $\mathcal{L}$ *sallittuja merkkijonoja* ovat esimerkiksi:

★★★★★★★★◇◇◇

★★★★◇◇◇◇

◇◇

★★★

Seuraavat jonot eivät ole *sallittuja merkkijonoja* :

★★★◇◇◇★

◇◇◇★

HUOM: Kieliopin (tulkinnan) yksiselitteisyys vrs. sen kuvaamisessa käytetty kieli?

STANDARDI SYNTAKTINEN METAKIELI SSM:

- kieliopin kuvaamiseen tarvitaan yksiselitteisempi, *formaalimpi* metakieli
- kieliopin kuvaaminen joukolla sääntöjä, jotka määrittävät ja nimeävät *sallitut merkkijonot* eli kielen sallitut rakenteet

> SSM-rakenteen syntaksi koostuu sille annettavasta nimestä, jota seuraa =-merkki, jonka jälkeen annetaan kokonaisuuden määrittelysääntö ja päätetään määritys ;-merkkiin. Kielen aakkostoon kuuluvat symbolit varustetaan “ ”-merkeillä. Kun määrittelysääntö voidaan muodostaa useammalla, toisilleen vaihtoehtoisella tavalla, erotetaan eri tapaukset toisistaan |-merkin avulla.

- aakkoston luetteleminen:

  luku = “0” | “1” | “2” | “3” | “4” | “5” | “6” | “7” | “8” | “9”;

  hexaluku = *luku* | “A” | “B” | “C” | “D” | “E” | “F”;

- *sallitun merkkijonon* järjestyksen määrääminen:

  kaksinumeroinen luku = *luku*, *luku*;

- . . . ja näiden yhdistelmä (eli aakkostona = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, .})

etumerkitön desimaaliluku	=	*etumerkitön kokonaisluku* | *desimaaliosa* | *etumerkitön kokonaisluku*, *desimaaliosa*;
desimaaliosa	=	“.”, *etumerkitön kokonaisluku*;
etumerkitön kokonaisluku	=	*luku* | *luku*, *etumerkitön kokonaisluku*;
luku	=	“0” | “1” | “2” | “3” | “4” | “5” | “6” | “7” | “8” | “9”;

EBNF:

- lähtökohtana Backus-Naur -formalismi (BNF) ALGOL 68 -kielen kuvaamiseksi
- kuvauksen rakennuspalikoina terminaali(symboli)t (`'nimi'`), nonterminaali(symboli)t (`< nimi >`), alkusymboli ja joukko kielioppisääntöjä eli produktioita, jotka ovat muotoa

  nonterminaali ::= *määrittely*

- määrittelyn syntaksi ja sen tulkinta:

Symboli	**Merkitys**
xy	määrittelyt x ja y peräkkäin (ilman pilkkua, vrt. SSM)
$[x]$	0 tai 1 määrittelyn x esiintymää
$\{x\}$	0 tai useampi määrittelyn x esiintymää
$x \mid y$	joko x tai y

EBNF in EBNF

```
< produktio >  ::= < nonterminaali > '::=' < jono > '.'

< jono >       ::= < alkio > | < alkio > < jono >

< alkio >      ::= < terminaali > | < nonterminaali > |
                    < kooste > | < rakenne >

< kooste >     ::= '(' < jono > ')'

< rakenne >    ::= < valinta > | < toisto > | < vaihtoehto >

< valinta >    ::= '[' < jono > ']'

< toisto >     ::= '{' < jono > '}'

< vaihtoehto > ::= < alkio > '|' < alkio >
```

SEMANTIIKKA:

- kielen jokaisen *sallitun merkkijonon* merkityksen määrittäminen
  - aakkoston symboleihin liitetään sovellusalueen konkreettinen arvo (“0” ei ole 0 ennenkuin siitä on sovittu!)
  - symboleista muodostettujen *sallittujen merkkijonojen* varsinainen merkitys
- tämäkin voitaisiin/pitäisi tehdä formaalisti, mutta . . .

Tulkinta $\mathcal{I}_1$ aikaisemmassa esimerkissä kuvatulle kielelle. Olkoot

$\bigstar$= 5

$\diamond$= 1

Symbolien asettaminen peräkkäin $\simeq$ lukujen laskeminen yhteen

$\diamond\diamond\diamond$ = 1 + 1 + 1 = 3 ,

$\bigstar\bigstar\bigstar\diamond\diamond$ = 5 + 5 + 5 + 1 +1 = 17 .

$\mathcal{I}_1$ ($\diamond\diamond\diamond$) = 3

Tulkinnalla $sallituille\ merkkijonoille$ arvoja joukosta $\{1, 2, 3, \ldots\}$

Tulkinta $\mathcal{I}_2$. Olkoot

$\mathcal{I}_2$ ($\bigstar$) = 10

$\mathcal{I}_2$ ($\diamond$) = 2

symbolien asettaminen peräkkäin $\simeq$ lukujen laskeminen yhteen
$\simeq \mathcal{I}_2(xy) = \mathcal{I}_2(x) + \mathcal{I}_2(y)$

$\mathcal{I}_2$ ($\diamond\diamond\diamond$) = $\mathcal{I}_2$ ($\diamond$) + $\mathcal{I}_2$ ($\diamond\diamond$) = $\mathcal{I}_2$ ($\diamond$) + $\mathcal{I}_2$ ($\diamond$) + $\mathcal{I}_2$ ($\diamond$) = 2 + 2 + 2 = 6 ,

$\mathcal{I}_2$ ($\bigstar\bigstar\bigstar\diamond\diamond$) = 10 + 10 + 10 + 2 +2 = 34 .

Tulkinnalla $sallituille\ merkkijonoille$ arvoja joukosta $\{2, 4, 6, \ldots\}$

PÄÄTTELYMEKANISMI:

Aksioomat: “riippumattomat” *sallitut merkkijonot* , jotka edustavat “erillisiä“, *sovellusalueen* “alkuperäisiä“ totuuksia

Päättelysäännöt: sääntöjä, joiden avulla aksioomista voidaan muodostaa niiden *välittöminä seurauksina* (immediate consequences) uusia *sallittuja merkkijonoja*

ESIMERKKI:

Aakkosto = { $*$, $\diamond$, $\circ$ }

Kielioppisäännöt:

lauseke = *tähtijono* ,” $\diamond$ ”, *tähtijono*, ” $\circ$ ”, *tähtijono*;
tähtijono = ” $*$ ” | *tähtijono*, ” $*$ ”;

Päättelymekanismi:

Aksiooma $* \diamond * \circ * *$

Sääntö “Jos $a \diamond b \circ c$ on *sallittu merkkijono* , missä a, b, c ovat tyyppiä *tähtijono*, niin $a \diamond b * \circ c *$ on tämän välitön seuraus.

Lauseke $* \diamond * * * * \circ * * * * *$ on *sallitun merkkijonon* $* \diamond * * * \circ * * * *$ välitön seuraus. Tämä nähdään, kun identifioidaan päättelysäännön sisältämät lausekkeet

$$\overbrace{*}^{a} \diamond \overbrace{* * *}^{b} \circ \overbrace{* * * *}^{c},$$

ja sovelletaan sääntöä “ilman aivoja” (tämän tietokone osaisi tehdä paremmin kuin me)

$$\overbrace{*}^{a} \diamond \overbrace{* * * *}^{b*} \circ \overbrace{* * * * *}^{c*},$$

ESIMERKKI (cont.):

Edellisen formaalin systeemin (yksi) tulkinta $\mathcal{I}$:

- Olkoon $* = 1$, $** = 2$, $*** = 3$, ... eli “formaalimmin”, induktiivisesti esitettynä

  $\mathcal{I}(*) = 1 \quad (* = 1)$,

  $\mathcal{I}(a*) = \mathcal{I}(a) + \mathcal{I}(*) \quad$ (n + 1 = (n) + 1 = ((n-1) + 1) + 1 = ...),

  missä a on tyyppiä *tähtijono*.

- Tarkoittakoon $\diamond$ normaalia yhteenlaskumerkkiä +
- Tarkoittakoon $\circ$ normaalia yhtäsuuruusmerkkiä =

Tässä semantiikassa formaalin kielen *tähtijonot* ovat kokonaislukuja ja kielen lausekkeet tulkitaan sallittujen merkkijonojen yhteenlaskuna

$$a + b = c,$$

joka siis normaalin aritmetiikan suhteen voi olla joko *tosi* tai *epätosi* sen suhteen päteekö saatu laskukaava. Kielen aksiooma tarkoittaa luonnollisesti perustapausta

$$1 + 1 = 2,$$

josta päättelysäännön avulla voidaan johtaa uusia, muotoa

$$a + (b + 1) = (c + 1)$$

olevia välittömiä seurauksia. Lopuksi on syytä muistuttaa siitä, että tässä esitetty tulkinta kuulosti järkevältä kun erityisesti symboliksi $\diamond$ valittiin ’+’-merkki. Formaalina systeeminä ihan “yhtä hyvä“ olisi saatu aikaan myös valitsemalla $\diamond$ edustamaan ’-’-merkkiä, mutta systeemin järkevä tulkinta olisi taatusti ollut paljon hankalampaa.

Todistukset ja lauseet:

- Formaaliin systeemiin $\mathcal{F}$ liittyvä *todistus* (proof) tarkoittaa *sallittujen merkkijonojen* äärellistä yhdistelmää, joista jokainen *sallittu merkkijono* on joko aksiooma tai annettujen päättelysääntöjen avulla muodostettu aksiooman tai *sallitun merkkijonon* välitön seuraus.
- Formaalin systeemin $\mathcal{F}$ *sallittu merkkijono* , joka voidaan todistaa tällä tavalla, on $\mathcal{F}$:n *lause* (theorem)
- erityisesti jokainen aksiooma sellaisenaan muodostaa $\mathcal{F}$:n lauseen

Esim. aikaisemmalle esimerkille pätee seuraava lause:

Lause 1. $* \diamond * * * \circ * * * *$

Todistus.

1	$* \diamond * \circ * *$	aksiooma
2	$* \diamond * * \circ * * *$	päättelysääntö sovellettuna 1:een
3	$* \diamond * * * \circ * * * *$	päättelysääntö sovellettuna 2:een
4	$* \diamond * * * * \circ * * * * *$	päättelysääntö sovellettuna 3:een

- huomataan, että todistuksen riveiltä 2 ja 3 saadaan uusia lauseita
- yhteenlaskutulkinnan mukaan lause vaikuttaa järkevältä (1 + 4 = 5)

Formaalin systeemin $\mathcal{F}$ tulkinta $\mathcal{I}$, jossa jokaiseen *sallittuun merkkijonoon* voidaan liittää totuusarvo **T** tai **E**, on

johdonmukainen, jos $\mathcal{F}$:n jokainen lause on tulkinnan suhteen **T**

täydellinen, jos tulkinnan suhteen jokainen **T** voidaan todistaa $\mathcal{F}$:n lauseeksi

esimerkiksi lauseke

$$* * * \diamond * * * * \circ * * * * * * *$$

on esimerkkikielen *sallittu merkkijono* , mutta sitä ei voida todistaa lauseeksi (eli esimerkkisysteemi on epätäydellinen)
$\Rightarrow$ täydentäminen, mutta johdonmukaisuus säilyttäen . . .

Johdot:

- todistuksien avulla formaaliin systeemiin voidaan muodostaa uusia totuuksia eli *sallittuja merkkijonoja*
- *johtojen* (derivations) avulla tehdään sama mutta lähtien liikkeelle muista kuin aksioomista
- *premissit* (premises), joita merkitään $\mathcal{P}$:llä, kuvaavat systeemin kyseisellä hetkellä jo sisältämiä *sallittuja merkkijonoja*
  $\Rightarrow$ *sallitun merkkijonon* $\mathcal{W}$ *johtamisella* tarkoitetaan $\mathcal{W}$:hen päättyvää äärellistä ketjua *sallittuja merkkijonoja* , joista jokaisen sisältämä lauseke on joko
  - $\mathcal{F}$:n aksiooma
  - $\mathcal{F}$:n premissi
  - ketjussa aikaisemmin esiintyneiden *sallittujen merkkijonojen* välitön seuraus, joka voidaan muodostaa annettujen päättelysääntöjen avulla
- ketju muodostaa tyyppiä “jos ..., niin ...” olevan rakenteen
- jos $\mathcal{W}$ voidaan johtaa premisseistä $\mathcal{P}$, merkitään tätä $\mathcal{P} \vdash \mathcal{W}$, missä metasymboli $\vdash$ tarkoittaa päättelysääntöjen mukaista loogista seurausta (syntactic turnstile)
- $\mathcal{P} \dashv\vdash \mathcal{W}$ tarkoittaa sitä, että sekä $\mathcal{P} \vdash \mathcal{W}$ että $\mathcal{P} \vdash \mathcal{W}$ pätevät;
  $\vdash \mathcal{W}$ pätee tyhjälle joukolle premissejä (eli jo todistettu lause tai aksiooma)
- siirrettäessä johdoissa käytetyt premissit jonkin <u>uuden</u> formaalin systeemin aksioomiksi siirtyvät vastaavasti kaikki johdot kyseisen systeemin lauseiksi

Esimerkki:

Johto 1. Aikaisemman esimerkin formaalille systeemille pätee seuraava tulos:

$$* * * \diamond * * \circ * * * * * \vdash * * * \diamond * * * * \circ * * * * * * *$$

Johtaminen.

1	$* * * \diamond * * \circ * * * * *$	premissi
2	$* * * \diamond * * * \circ * * * * * *$	1 & **sääntö**
3	$* * * \diamond * * * * \circ * * * * * * *$	2 & **sääntö**

$\square$