JOHDANTO KURSSIN AIHEPIIRIIN

1. Näkemyksiä loogiseen päättelyyn
2. Logiikan soveluskohteita tietojenkäsittelyssä
3. Joitain esitietoja

1 Näkemyksiä loogiseen päättelyyn

- **Inhimillinen päättely:**
  Esim. ihmisen suorittama syiden ja seurausten analysointi.
  Keskeisiä piirteitä: päättelykyvyn ja
  päättelyyn käytettävissä olevan rajallisuus.
- **Formaali päättely:**
  Matemaattinen logiika tekee loogisesta päättelystä formaalin:
  - väittämät esitetään formaalilla kielellä ja
  - johtopäätösten hyväksyttävyydelle annetaan eksaktit kriteerit.

  Malli on ideaalinen (vrt. inhimillinen päättely) ja abstrakti.
  Päättely voidaan palauttaa merkkijonojen käsittelyksi ao. kielessä.
- **Automaattinen/mekaaninen päättely**
  Toteutetaan looginen päättely tietokoneohjelmana (päättelykone).

Esimerkki. Tarkastellaan yksinkertaisen hälyttimen kuvausta:

Järjestelmän kuvaus:
Jos savua, niin hälytys.
Jos ovi auki, niin hälytys.
Havainnot: savua tai ovi auki.

$\Longrightarrow$

Formaali esitys:
$A \rightarrow C$
$B \rightarrow C$
$A \vee B$

$\Downarrow$

Päättelykone

$\Downarrow$

Johtopäätös: C

$\Longleftarrow$

Tulkinta järjestelmässä:
Hälytys!

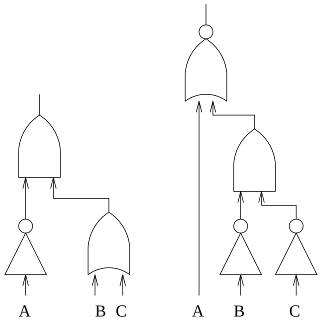
2 Logiikan soveluskohteita tietojenkäsittelyssä

Voidaan nähdä karkea kahtiajako:

- **Järjestelmän ominaisuuksien määrittely ja analysointi: esim.**
  - Ohjelman ehtolausekeiden muokaaminen
  - Ohjelmien vaatimusmäärittelyt ja synteesi
  - Järjestelmien oikeellisuustarkastelut
- **Päättelykomponentti järjestelmän osana: esim.**
  - Ehtolausekeiden evaluointi (mm. ohjelmointikielet)
  - Loogisten lausekeiden tulkitseminen käskyiksi tietokoneelle
  - Sääntöpohjainen päättely (mm. asiantuntijajärjestelmät)
  - Rajoiteohjelmointi

Esimerkki. Kombinatoriset piirit

Laskevatko seuraavat kombinatoriset piirit samat funktiot?



Piirien kuvaukset lauselogiikalla: $\neg A \wedge (B \vee C)$ ja $\neg(A \vee (\neg B \wedge \neg C))$

Esimerkki. Ohjelmien ehtolausekeet

Olkoon `myptr` tyyppiä “`char *`” C-kielessä.

```
/* TAPA 1 */

if(myptr != NULL && myptr[0] == '/') doit(myptr);
if(!(myptr == NULL || myptr[0] == '.')) dothat(myptr);

/* TAPA 2 */

if(myptr != NULL) {
  if(myptr[0] == '/') doit(myptr);
  if(myptr[0] != '.') dothat(myptr);
}
```


Esimerkki. Ohjelmankehitys

Tiedetään: jos $X_1 < 0$ tai $X_2 < 0$ tai ... tai $X_n < 0$, niin $Z < 0$.

Vertaa:

```
if (Z < 0)                                if (Z < 0)
    do_0()                                    do_0();
else {
    if (X1 < 0)
        do_1();
    if (X2 < 0)
        do_2();
    ...
    if (Xn < 0)
        do_n();
}
```


Esimerkki. Ohjelmien oikeellisuustarkastelut

- **Induktio** tietorakenteiden suhteen

  ```
  duplicate([],[]).
  duplicate([E|R1],[E|[E|R2]]) :- duplicate(R1,R2).
  ```

  Voidaan osoittaa *rakenteisella induktiolla*, että proseduuri `duplicate` käsittelee oikein mielivaltaisen pitkiä listoja.

- **Induktio** silmukan suorituskertojen suhteen

  ```
  int loop() { int n = 0; while(n<1000) n++; return n; }
  ```

  Voidaan osoittaa, että $\mathrm{n} \leq 1000$ on while-silmukan *invariantti* (eli ominaisuus, joka säilyy silmukaa suoritettaessa).

Esimerkki. Relaatiotietokannat

Predikaattilogiikalla voidaan kuvata relaatiotietokantojen primitiivit:

- Relaatioiden R_1 ja R_2 unioni: $\forall\overline{x}(R_1(\overline{x}) \vee R_2(\overline{x}) \rightarrow P(\overline{x}))$.
- Relaatioiden R_1 ja R_2 leikaus: $\forall\ \overline{x}(R_1(\overline{x}) \wedge R_2(\overline{x}) \rightarrow P(\overline{x}))$.
- Relaation R projektio: esim. $\forall x \forall y \forall z(R(x,y,z) \rightarrow P(x,z))$.
- Relaatioiden R_1 ja R_2 luonnollinen yhdiste (alla y:n suhteen):

$$\forall x \forall y \forall z(R_1(x,y) \wedge R_2(y,z) \rightarrow R(x,y,z)).$$

Esimerkki. Deduktiiviset tietokannat

linkki(otaniemi, tapiola)

linkki(otaniemi, lehtisaari)

$\forall x \forall y$(linkki$(x,y) \rightarrow$ linkki(y,x))

$\forall x \forall y$(linkki$(x,y) \rightarrow$ yhteys(x,y))

$\forall x \forall y \forall z$(yhteys$(x,y) \wedge$ yhteys$(y,z) \rightarrow$ yhteys(x,z))

Kuinka selvitetään vastaus kyselyyn yhteys(tapiola, lehtisaari)?
Entä perinteisellä relaatiotietokannalla (SQL-kysely)?

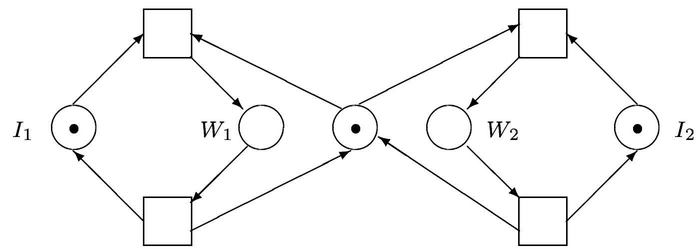
Esimerkki. Logiikkaohjelmointi (PROLOG)

```
append([],L,L).
append([A|T],L,[A|S]) :- append(T,L,S).

?- append([1,2,3],[4,5,6],X).

X = [1,2,3,4,5,6]

?- append([1,X,3],Y,[1,2,3,4]).

X = 2, Y=[4]
```

Vrt. yleinen näkemys: PROGRAM = LOGIC + CONTROL

Esimerkki. Loogiset määrittely- ja kyselykielet

Mallinnetaan *semafori* Petri-verkkojen (automaattien yleistys) avulla.



- I_i tarkoittaa, että prosessi i on toimettomana, ja W_i tarkoittaa, että prosessi i kirjoittaa prosessien yhteiseen muistipaikkaan.
- Voimme todeta, että lause $I_1 \vee I_2$ on aina tosi ja että lause $W_1 \wedge W_2$ on aina epätosi järjestelmän *saavutettavissa* tiloissa.

Esimerkki. Loogisen kielen laajennuksia

Tyypillisiä laajennuksia ovat modaalilogiikat, kuten esim.

- Aikalogiikka

  Tyypillisiä operaattoreita $\Box$ (aina), $\Diamond$ (joskus) ja $\bigcirc$ (seuraavalla ajanhetkellä)

  Turvallisuusominaisuudet: $\Box(P \vee Q)$

  Reiluusominaisuudet: $\Diamond(P \wedge \bigcirc R)$

- Tietämys- ja uskomuslogiikat

  Operaattoreina mm. $\mathbf{K}$ (tietää) ja $\mathbf{B}$ (uskoa).

  Esim. $\mathbf{K}P \rightarrow \mathbf{K}\mathbf{K}P$ (itsetutkiskelu)

  $\neg\mathbf{B}P \rightarrow \neg P$ (suljetun maailman oletus)

3 Joitain esitietoja

3.1 Induktioperiaate luonnollisille luvuile

Luonnollisten lukujen joukko määritellään induktiivisesti:

- 0 on luonnollinen luku ja
- jos n on luonnollinen luku, niin myös $n+1$ on luonnollinen luku.

Jos halutaan osoittaa, että kaikilla luonnollisilla luvuilla n on jokin ominaisuus P, sovelletaan seuraavaa periaatettа:

Määritelmä. Induktioperiaate.
Olkoon $P(n)$ luonnolliselle luvulle n määritelty ominaisuus.
Jos $P(0)$ ja $\forall n \in \mathbb{N} : (P(n) \rightarrow P(n+1))$, niin $\forall n \in \mathbb{N} : P(n)$.

Esimerkki. Todistetaan, että kaikille luonnollisille luvuille n pätee

$$2^0 + 2^1 + \ldots + 2^n = 2^{n+1} - 1.$$

Perustapaus: $2^0 = 1$ ja $2^1 - 1 = 2 - 1 = 1$.

Induktiohypoteesi: $2^0 + 2^1 + \ldots + 2^{n-1} = 2^n - 1$

Induktioaskel:
$$\begin{aligned} 2^0 + 2^1 + \ldots + 2^n &= (2^0 + 2^1 + \ldots + 2^{n-1}) + 2^n \\ &= (2^n - 1) + 2^n \\ &= 2 \times 2^n - 1 = 2^{n+1} - 1 \end{aligned}$$

Tuloksen tietojenkäsittelyllinen merkitys: täydellisessä binääripuussa on lehtisolmuja yksi enemmän kuin sisäsolmuja.

- Jos kysymyksessä on hakupuu, niin puun syvyyden n kasvattaminen $n+1$:een johtaa työmäärän kaksinkertaistumiseen.

Induktioperiaatteesta käytetään usein tiettyjä muunnelmia.

- **Täydellinen induktio** (induktio-oletusta vahvennettu):

  Jos $P(0)$ ja
  $\forall n \in \mathbb{N} : (\forall m \in \mathbb{N} : (m < n+1 \rightarrow P(m)) \rightarrow P(n+1))$,
  niin $\forall n \in \mathbb{N} : P(n)$.

Esimerkki. Jokainen luonnollinen luku $n > 1$ voidaan kirjoittaa alkulukujen tuloksi.

- **Yhtäaikainen induktio k:n eri ominaisuuden $P_1, \ldots, P_k$ suhteen:**

  Jos $P_1(0), \ldots, P_k(0)$ ja
  $\forall n \in \mathbb{N} : \forall i \in \{1, \ldots, k\} : (\forall j \in \{1, \ldots, k\} : P_j(n) \rightarrow P_i(n+1))$,
  niin $\forall n \in \mathbb{N} : P_1(n), \ldots, \forall n \in \mathbb{N} : P_k(n)$.

  Käyttökelpoinen tilanteissa, joissa $P_1, \ldots, P_k$ riippuvat toisistaan.

3.2 Joukko-opin peruskäsitteet

Tarkastellaan joukkoja $A = \{a, b\}$ ja $B = \{b, c\}$.

- Joukon jäsenyys: $a \in A$, $b \in A$ ja $c \notin A$
- Joukkojen yhtäsuuruus: $A \neq B$
  (koska A:lla ja B:llä ei ole samat alkiot).
- Tyhjä joukko: $\emptyset$ (tai vaihtoehtoisesti $\{\}$)
- Osajoukko: $\emptyset \subseteq A$, $A \not\subseteq B$ ja $B \not\subseteq A$.
- Aito osajoukko: $\emptyset \subset A$
- Joukkojen kardinaliteetti: $|\emptyset| = 0$ ja $|A| = |B| = 2$

Keskeisiä joukkojen välisiä operaatioita

Tarkastellaan edelleen joukkoja $A = \{a, b\}$ ja $B = \{b, c\}$:

- Joukkojen unioni: $A \cup B = \{a, b, c\}$
- Joukkojen leikaus: $A \cap B = \{b\}$
- Joukkojen erotus: $A - B = \{a\}$ ja $B - A = \{c\}$
- Karteesinen tulo: $A \times B = \{\langle a, b\rangle, \langle a, c\rangle, \langle b, b\rangle, \langle b, c\rangle\}$
- Potenssijoukko (eli joukon kaikki osajoukot):

$$\mathbf{2}^A = \{\emptyset, \{a\}, \{b\}, \{a, b\}\} \text{ ja } |\mathbf{2}^A| = 2^{|A|} = 2^2 = 4.$$

Huomio. Joukko $\mathbf{2}^{\emptyset} = \{\emptyset\}$ eli joukko, jonka alkiona on tyhjä joukko!

3.3 Relaatiot

- Joukkojen $A_1, \ldots, A_n$ *karteesinen tulo*
  $A_1 \times \cdots \times A_n = \{\langle a_1, \ldots, a_n\rangle \mid a_1 \in A_1, \ldots, a_n \in A_n\}$.
- Jos $A_1 = A$, ..., $A_n = A$, saadaan
  $A^n = \underbrace{A \times \cdots \times A}_{n \text{ kpl}} = \{\langle a_1, \ldots, a_n\rangle \mid a_1 \in A, \ldots, a_n \in A\}$.
  Erikoistapaukset: $A^1 = \{\langle a\rangle \mid a \in A\} = A$ ja $A^0 = \{\langle\rangle\}$.
- Joukkojen $A_1, \ldots, A_n$ välinen n-paikkainen *relaatio* R on karteesisen tulon $A_1 \times \cdots \times A_n$ osajoukko.

Esimerkki. Kaksi relaatiota luonnollisten lukujen välillä:

$\{n \mid n \in \mathbb{N} \text{ on pariton}\} \subseteq \mathbb{N}^1$

$\{\langle x, y, z\rangle \mid x^2 + y^2 = z^2,\ x \in \mathbb{N},\ y \in \mathbb{N},\ z \in \mathbb{N}\} \subseteq \mathbb{N}^3$

Keskeisiä relaatioiden ominaisuuksia

Binäärirelaatio $R \subseteq A \times A$ on

- *refleksiivinen*, joss kaikille $a \in A$ pätee $\langle a, a\rangle \in R$.
- *irrefleksiivinen*, joss kaikille $a \in A$ pätee $\langle a, a\rangle \notin R$.
- *symmetrinen*, joss kaikille $a \in A$ ja $b \in A$ pätee:

  jos $\langle a, b\rangle \in R$, niin $\langle b, a\rangle \in R$.
- *asymmetrinen*, joss kaikille $a \in A$ ja $b \in A$ pätee:

  jos $\langle a, b\rangle \in R$, niin $\langle b, a\rangle \notin R$.
- *transitiivinen*, joss kaikille $a \in A$, $b \in A$ ja $c \in A$ pätee:

  jos $\langle a, b\rangle \in R$ ja $\langle b, c\rangle \in R$, niin $\langle a, c\rangle \in R$.
- *ekvivalenssirelaatio*, joss R on refleksiivinen, symmetrinen ja transitiivinen.

3.4 Funktiot

Funktio $f : A_1 \times \cdots \times A_n \to A$ on relaatio $f \subseteq A_1 \times \cdots \times A_n \times A$, joka toteuttaa *funktionaalisuusehdon*:

kaikille $a_1 \in A_1$, ..., $a_n \in A_n$ on olemassa täsmälleen yksi $a \in A$ siten, että $\langle a_1, \ldots, a_n, a\rangle \in f$ (eli f:n arvo pisteessä $\langle a_1, \ldots, a_n\rangle$).

Kyseistä alkiota merkitään lausekkeella $f(a_1, \ldots, a_n)$.

Keskeisiä funktioiden ominaisuuksia

Funktio $f : A \to B$ on

- *injektio*, joss kaikille $a \in A$ ja $b \in A$ pätee $f(a) \neq f(b)$.
- *surjektio*, joss kaikille $b \in B$ on olemassa $a \in A$ siten, että $f(a) = b$.
- *bijektio*, joss f on sekä injektio että surjektio.