



# ***Teoreettisen tutkimuksen raportoinnista Tutki ja kirjoita -kurssi, s-2005***

Pekka Kilpeläinen

Kuopion yliopisto

Tietojenkäsittelytieteen laitos

## Algoritmisten artikkelien sisällöstä, esitystavoista ja rakenteesta

- ⑥ Algoritmien esittämisestä
  - △ mitä ja miten?
- ⑥ Tieteellisen kirjoitelman rakenteesta
  - △ funktionaalisuus ja sidosteisuus
- ⑥ Esimerkki
  - △ algoritmisen artikkelin sisällöstä ja rakenteesta

# Miten esitellä algoritmisia tuloksia?

Luku 7 (Algorithms) teoksessa J. Zobel: *Writing for Computer Science, Second Edition*. Springer-Verlag, 2004. Ks. myös luku 5 (Mathematics)

Pelkkä algoritmi ei riitä; Esiteltävä

- ⑥ selittäen
- ⑥ perustellen merkittävyys, oikeellisuus, tehokkuus

Kontribuutio sekä sen perustelut:

Tulosten merkitys ja paikkansapitävyys

# Mitä kertoa algoritmista?

(Zobel, sekä Nevalainen raportissa Järvinen&Järvinen: Tutkimustyön metodeista. Opinpaja Oy, Tampere, 1996.)

Tilanteesta riippuen osa tai kaikki allaolevista:

1. Algoritmin idea (sanallisesti, korkealla tasolla);  
sovellusala, mahdolliset rajoitukset
2. Syötteet, tulokset, keskeiset tietorakenteet
3. Algoritmin kuvaus
4. Havainnollistava esimerkki
5. Oikeellisuuden perusteluun mahd. tarvittavat ominaisuudet;  
Oikeellisuustodistus
6. Tehokkuustarkastelu (asymptoottinen analyysi ja/tai koejärjestely)
7. Toteutuskoodi (?)

# Algoritmien esitystapoja

Miten esittää algoritmeja?

- ⑥ Sopivalla abstraktiotasolla
- ⑥ Asiantuntijaa silmällä pitäen  
(EI konetta tai 1. vuoden opiskelijaa)

**Vertaa**  $d \leftarrow \max\{D_{i,j} \mid 1 \leq i \leq m, 1 \leq j \leq n\}$   
ja

$d := D[1, 1];$

**for**  $i := 1, \dots, m$  **do**

**for**  $j := 1, \dots, n$  **do**

**if**  $d < D[i, j]$  **then**  $d := D[i, j];$

# Algoritmien esitystyylejä

**Numeroitu askelluettelo, esim:**

⋮

2. Muodosta toisen tason hajautusfunktio:

2.1 Valitse satunnainen  $h \in \mathcal{H}_{a,b}$ ;

2.2 Jos  $h(x) = h(y)$  joillain  $x, y \in K$ ,  $x \neq y$ ,  
palaa vaiheeseen 2.1;

Sanallista selittämistä tukeva

Algoritmin rakenne ( $\rightarrow$  toteutus) saattaa hämärtyä

# Algoritmien esitystyylejä (2)

## Pseudokoodi

- ⑥ (ollut) oma suosikki
- ⑥ Tukee toteutusta
  - saattaa olla liian detaljoitu

## Muita

“proosakoodi”, “kirjallinen koodi” (literate code)

ohjelmointikieletkin joskus

Vuokaavioita en suosittele vaativien algoritmien esittämiseen

# Algoritmien $\text{\LaTeX}$ oinnista

Algoritmipaketit kuten `algorithmicx` mahdollisesti hyödyllisiä (Ks. esim. Typesetting pseudocode in LaTeX, UK TeX FAQ)

- ⑥ siisti yhtenäinen tyyli
- ⑥ automaattinen rivien numerointi ja sisennys

Olen kuitenkin nojautunut perus- $\text{\LaTeX}$ iin. Miksi?

- ⑥ laiskuus, ajanpuute ;-(
- ⑥ tietty muotoilun vapaus
- ⑥ työläys murto-osa algoritmiesityksen laatimisen vaativuudesta
- ⑥ siirtyminen abstraktimpiin kuvaustapoihin

# Esimerkkimäärittelyjä

## Tabulointiympäristö algoritmeille/ohjelmille:

```
\newenvironment{program}{\begin{tabbing}
  xxxx \= xxxx \= xxxx \= xxxx \= xxxx
\= xxxx \= xxxx \= xxxx \= xxxx \= xxxx \= \kill}
{\end{tabbing}}
```

## Komennot pseudokoodin avainsanoille:

```
\newcommand{\Forall}{\textbf{for all}}
\newcommand{\If}{\textbf{if}}
...
```

## Algoritmien numerointi:

```
\newtheorem{Algorithm}{Algorithm}[section]
```

# Käyttöesimerkki

```
\begin{Algorithm}\label{alg:gen-match}
General tree matching scheme.
\begin{description}
\item[Input:] Pattern tree  $P = (V, E, \text{juuri}\{P\})$ 
and target tree  $T = (W, F, \text{juuri}\{T\})$ .
\item[Output:] The nodes of  $W$  that are occurrences of  $P$ .
\item[Method:]
\end{description}
\begin{program}
\> \For\ $w := 1, \ldots, n$ \Do\
\> \Comment{Process the target nodes in postorder} \
\> \> $M(w) := \emptyset$; \
\> \> \Forall\ $v := 1, \ldots, m$ \Do\
\> \> \Comment{Process the pattern nodes in postorder} \
\> \> \> \If\ $\text{lab}\{v\} = \text{lab}\{w\}$ \Then \
...
\end{program}
\end{Algorithm}
```

# Ladonnan tulos

**Algorithm 4.3** General tree matching scheme.

**Input:** Pattern tree  $P = (V, E, \text{root}(P))$  and target tree  $T = (W, F, \text{root}(T))$ .

**Output:** The nodes of  $W$  that are occurrences of  $P$ .

**Method:**

**for**  $w := 1, \dots, n$  **do**

**comment:** Process the target nodes in postorder;

$M(w) := \emptyset$ ;

**for all**  $v := 1, \dots, m$  **do**

**comment:** Process the pattern nodes in postorder;

**if**  $\text{label}(v) = \text{label}(w)$  **then**

...

# Tekstin funktionaalisuus ja sidosteisuus

G.G. Gopen & J.A. Swan: The science of scientific writing. *American Scientist* (Nov–Dec 1990), Vol. 78, 550–558 ja L.A. Crowlin tiivistelmä siitä ynnä henkilökohtaisia ajatuksia

Huomioitavia lukijan odotuksia (jotta viesti välittyisi):

Kullakin *yksiköllä* (unit of discourse; lause, virke, kappale, luku, artikkeli) on **yksi funktio** ("pointti", "tarina") sekä **suunta**

alku → loppu

aihe/konteksti → tulos/väittämä/huipentuma

vanha/tunnettu → uusi/painotettu asia

*topic position ja stress position*

Looginen eteneminen suunnattujen yksikköjen mielekkäästä kytkeytymisestä

# Kontekstuaalisuus

Johdannossa, yhteenvedossa ja lukujen alussa yleensä artikkelin osia selittävää ja yhteen sitovaa ”liimatekstiä”

Tärkeä tuki lukijalle: osien merkitys ja rooli

Voi olla eksplisiittistä:

”The rest of this paper is organized as follows. In Section 2 we introduce . . .”

”We showed in Section 3 that [a specific problem can be solved efficiently] . . . On the other hand, [related problems] were shown NP-hard in Section 4.”

Muutoin yleensä parempi pyrkiä ”läpinäkyvyyteen”, s.o., keskittyä aiheeseen ja välttää huomion kiinnittämistä paperiin ja sen tekijöihin

# Artikkeliesimerkki

P. Kilpeläinen, H. Mannila:  
Ordered and unordered tree inclusion. *SIAM Journal on Computing* 24(2) (April 1995), 340–356).

- ⑥ Hyvä vai varoittava esimerkki? ;-)
- ⑥ Mitä kerrotaan algoritmisesta ongelmasta ja sen ratkaisusta
- ⑥ Miten?
- ⑥ Artikkelin yleinen rakenne