Berühmte Informatiker

Teil 10:

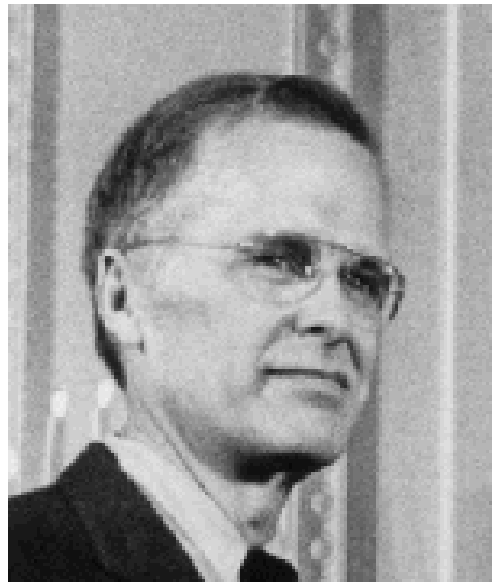
J. Backus & D. E. Knuth

1924- 1938-

John Backus

- * 03.12.1924 in Philadelphia

- Vorzeitiger Abbruch des Studiums der Chemie (1942) und der Medizin (1945)
- Während der Arbeit als Radiomechaniker entdeckt Backus seine Leidenschaft für Mathematik
- Studium der Mathematik an der Columbia University (Abschluss 1949)
- 1950-1991 Arbeit bei IBM



Höhepunkte der Karriere:

- Entwicklung von Fortran, der ersten High-Level Programmiersprache
- Entwicklung der Backus-Naur-Form zur eindeutigen Beschreibung der Syntax von Programmiersprachen

Auszeichnungen:

- 1977 ACM Turing Award für High-Level-Programmier-systeme und formale Verfahren zur Spezifikation von Programmiersprachen.

Fortran

- FORmula TRANslation: basiert auf dem Vorschlag von John Backus
- Programmiersprache, die insbesondere zur Berechnung numerischer Probleme eingesetzt wurde.
- Entwicklung von 1954-1957
- Auf Anregung von Backus war der Fortran-Compiler von Anfang an zu Optimierungen fähig.
- Programmierstil basierte auf der Goto-Anweisung.
- Zitat Backus:

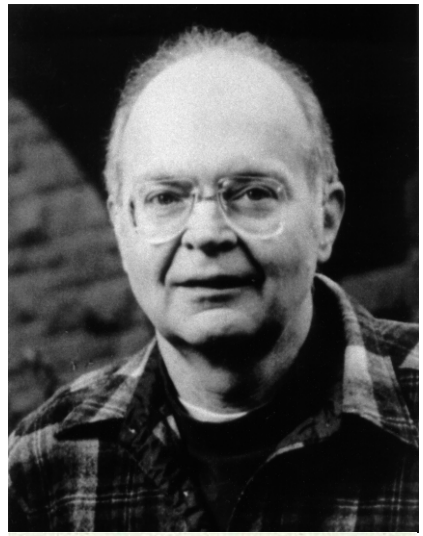
  „*We did not know what we wanted and how to do it. It just sort of grew. The first struggle was over what the language would look like. Then how to parse expressions - it was a big problem and what we did looks astonishingly clumsy now....*“

Backus-Naur-Form

- Zur exakten und eindeutigen Spezifikation von Programmiersprachen erfanden John Backus und Peter Naur die Backus-Naur-Form (BNF).
- Beispiel:

  <program> ::= program
  <declaration_sequence>
  begin
  <statements_sequence>
  end ;
- Verschiedene Änderungen und Erweiterungen der BNF wurden durchgeführt. Die heute gebräuchliche Version ist die extended Backus Naur Form (EBNF)

Donald Ervin Knuth



- * 10.01.1938 in Milwaukee

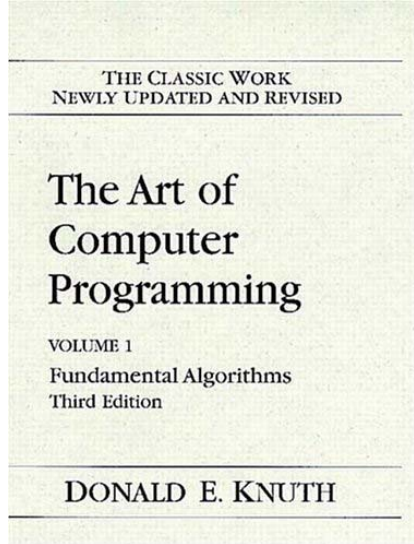
- Bachelor und Master 1960 an der Case Western Reserve University
- 1963 Ph.D. in Mathematik am California Institute of Technology
- 1963-1968 Mathematik Professor am California Institute of Technology
- seit 1968 Professor für Informatik an der Stanford University

Höhepunkte der Karriere:

- Verfasser der „Bibel für Informatiker“: The Art of Computer Programming
- Entwickler von Tex und Metafont

Auszeichnungen:

- 1974 ACM Turing Award für die Analyse von Algorithmen und den Entwurf von Programmiersprachen



Knuth-Morris-Pratt-Algorithmus

- Algorithmus zur Suche eines Musters (Pattern) in einem Text der die Anzahl der benötigte Vergleiche stark reduziert gegenüber dem naiven Ansatz.
- Naiver Ansatz:
  Gegeben: Text `text` der Länge *n*, Muster `pattern` der Länge *m*
  Ergebnis: Position im Text an der das Muster beginnt, sonst -1

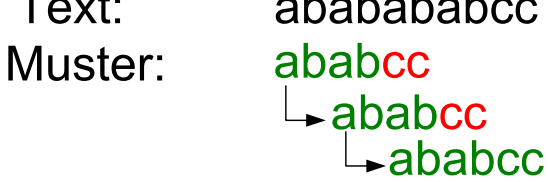
```
int naiveSearch(text,n,pattern,m){
    int i=0,j=0;
    while(i<n-m){
        j=0;
        while(text[i+j]==pattern[j]) j++;
        if(j==m) return i;
        i++;
    }
}
```

- Der naive Algorithmus vergißt all sein Wissen über die nötigen Vergleiche. Im schlechtesten Fall braucht der Algorithmus O(m*n) Vergleiche.

KMP-Algorithmus: Fortsetzung

Vorbemerkungen:

- Die Begriffe Präfix und Suffix haben wir bereits kennen gelernt.
- Ein Wort p ist ein echtes Präfix von w, falls p ein Präfix von w ist und |p|<|w| gilt. Die Definition eines echten Suffixes erfolgt analog.
- Ein Wort r ist ein Rand von w, falls r sowohl echtes Präfix, als auch echtes Suffix von w ist: ab ist Rand von abcab.
- Der KMP-Algorithmus untersucht zunächst das Muster und erarbeitet eine Tabelle, um wie viele Positionen die Suche nach rechts verschoben werden kann im Fall eines Mismatch.
- Beispiel:



Die Suchposition kann um |*p*|-|*r*| Positionen verschoben werden, wobei *p* das übereinstimmende Präfix des Muster und *r* der Rand von *p* ist.

- Eine gute Beschreibung findet sich unter http://www.iti.fh-flensburg.de/lang/algorithmen/pattern/kmp.htm

Books in Print by Donald E. Knuth

Click web links for current news about each book of interest. Lists of errors and amendments can be downloaded as plain TeX files or read from DVI files or PostScript files cited on the relevant web pages. You are entitled to a **reward** of at least $2.56 if you are the first person to report a bona-fide error not on those lists. Each page tells you how to report an error for the book in question.

The Art of Computer Programming (TAOCP)

The TeXbook

The METAFONTbook

Computers & Typesetting

Concrete Mathematics