Algoritma CYK

Teori Bahasa dan Automata

Semester Ganjil 2013

Jum'at, 06.12.2013

Dosen pengasuh:

Kurnia Saputra ST, M.Sc

Email: kurnia.saputra@gmail.com

Jurusan Informatika

Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Syiah Kuala

Algoritma CYK

Algoritma CYK adalah sebuah algoritma yang digunakan untuk membuktikan apakah sebuah word w di-generate oleh grammar context free atau tidak.

Algoritma CYK dikembangkan oleh John Cocke, Daniel Younger, dan Tadao Kasami.

Untuk dapat menggunakan algoritma ini dibutuhkan grammar context free G dalam bentuk Chomsky normal form, dimana word w adalah sebagai input, dan outputnya adalah sebuah pembuktian apakah word w merupakan bahasa dari grammar G atau bukan.

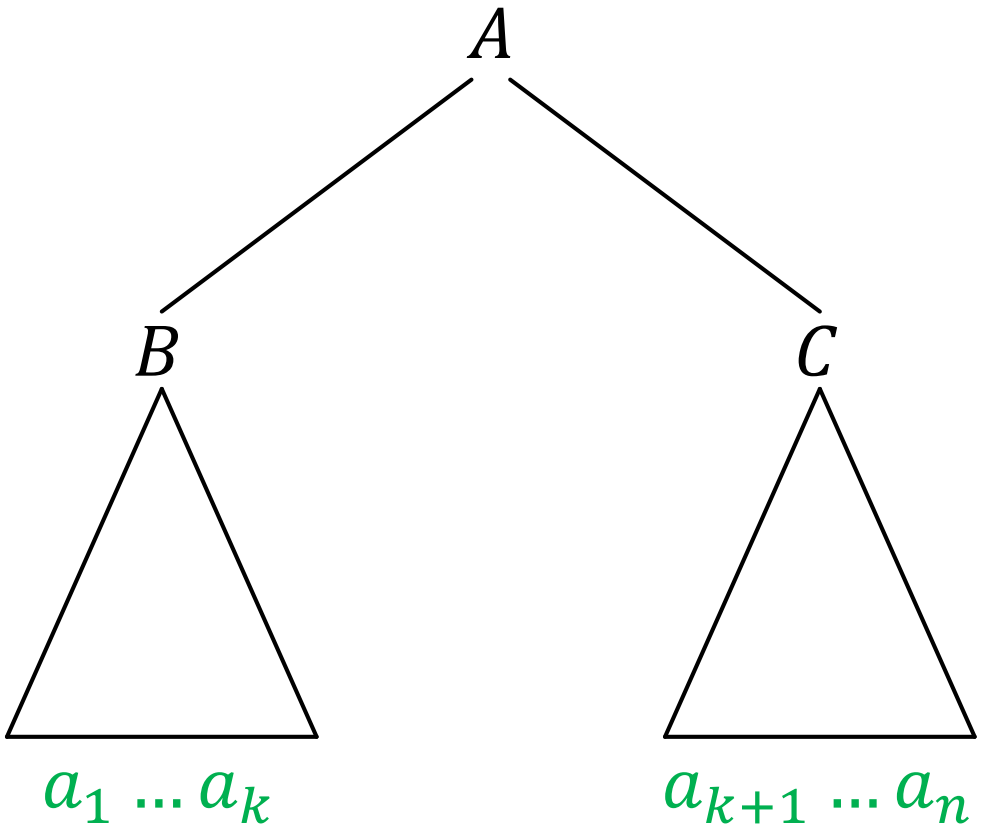
Algoritma CYK

Konsep: Diketahui word $w \in \Sigma^*$. Kita ingin mengetahui dari variabel apa saja word tersebut dapat di-derivasi.

- Kemungkinan 1: $x = a \in \Sigma$, dimana x terdiri dari simbol alphabet tunggal. Kemudian w hanya bisa di-derivasi dari variabel A dimana terdapat $A \longrightarrow a$.
- Kemungkinan 2: $x = a_1 \dots a_n$ dimana $n \geq 2$. Pada kasus ini production $A \longrightarrow BC$ harus dipilih terlebih dahulu, dimana ada satu bagian dari word $a_1 \dots a_k$ harus di-derivasi dari B dan satu bagian lagi dari word $a_{k+1} \dots a_n$ di-derivasi dari C $(1 \leq k < n)$.

Algoritma CYK

Kemungkinan 2 dapat digambarkan dengan skema sebagai berikut:



Algoritma CYK

Tapi masih belum begitu jelas bagaimana cara memisahkannya word x karena index k sangat besar.

Maka: Kita harus mencoba semua kemungkinan k.

Jika diketahui $x = a_1 \dots a_n$, dimana $1 < k < n$ lakukan langkah berikut:

- Cek apakah himpunan variabel V_1 dapat derivasi $a_1 \dots a_k$.
- Cek apakah himpunan variabel V_2 dapat derivasi $a_{k+1} \dots a_n$.
- Cek apakah variabel A, B, C dimana $(A \longrightarrow BC) \in P$, $B \in V_1$ dan $C \in V_2$. Dan x untuk kasus ini di-derivasi dari A.

Algoritma CYK

Untuk menghindari duplikasi, aplikasikan metode dynamic programming, artinya:

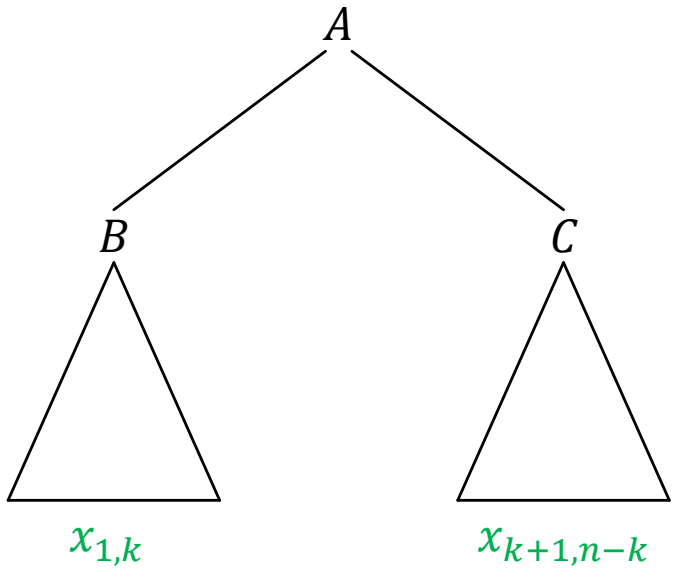
- Tentukan dahulu semua variabel yang bisa derivasi subword dengan panjang 1.
- Kemudian tentukan variabel yang bisa derivasi subword dengan panjang 2.
- …
- Terakhir, tentukan semua variabel dimana x bisa di-derivasi. Jika simbol start S berada diantara variabel, maka x adalah bahasa dari grammar tersebut.

Algoritma CYK

Notasi: $x_{i,j}$ menunjukkan bahwa subword x berada pada lokasi i dan memiliki panjang j.

$x = a_1 \dots a_n \quad \rightsquigarrow \quad x_{i,j} = a_i \dots a_{i+j-1}$

Dengan notasi tersebut, maka pohon variabel menjadi:



Algoritma CYK

$T_{i,j}$ adalah himpunan variabel dimana $x_{i,j}$ bisa di-derivasi.

$T_{i,j}$ ditentukan dari himpunan $T_{i',j'}$ dan $j' < j$, dimana:

$$T_{i,j} = \{A \mid (A \longrightarrow BC) \in P \text{ dan } k < j \text{ dimana } B \in T_{i,k} \text{ dan } C \in T_{i+k,j-k}\}$$

Algoritma CYK

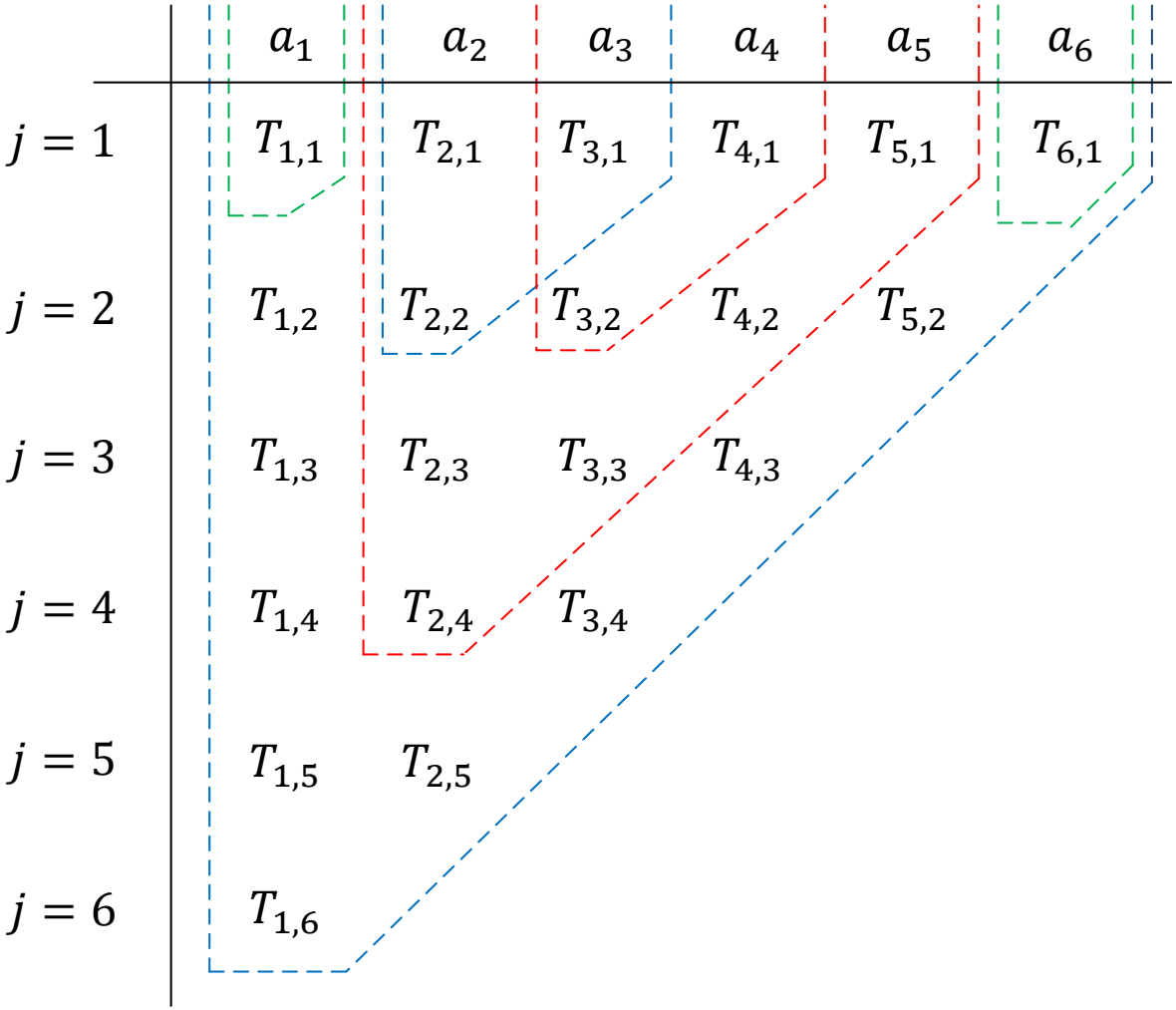
Eksekusi algoritma:

Masukkan himpunan variabel $T_{i,j}$ ke tabel:

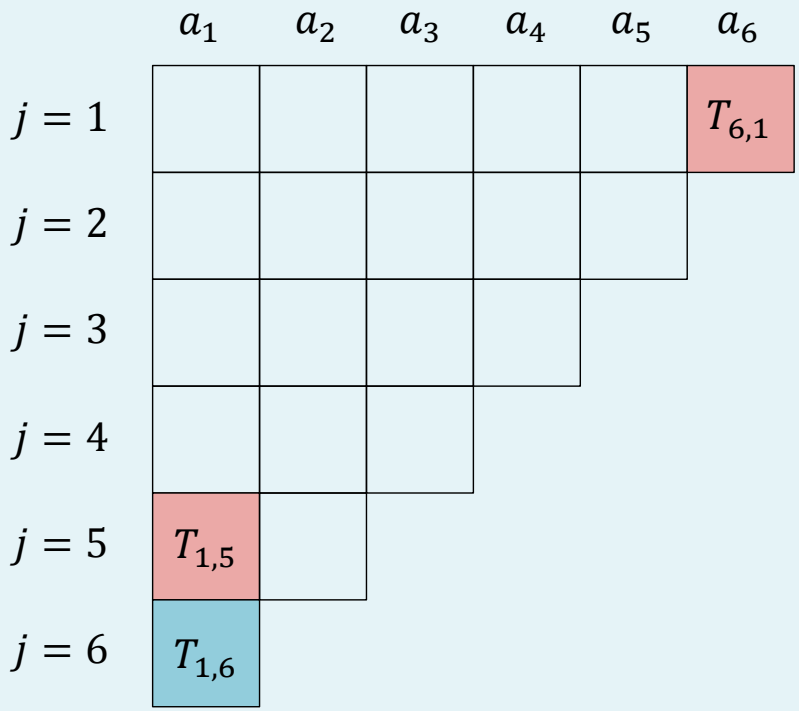
	a_1	a_2	…		a_{n-1}	a_n
$j = 1$	$T_{1,1}$	$T_{2,1}$	…	…	$T_{n-1,1}$	$T_{n,1}$
$j = 2$	$T_{1,2}$	$T_{2,2}$	…	…	$T_{n-1,2}$	
	…	…	…	…		
…	…	…	…			
$j = n - 1$	$T_{1,n-1}$	$T_{2,n-1}$				
$j = n$	$T_{1,n}$					

Algoritma CYK

Variabel akan derivasi subword sebagai berikut:



Algoritma CYK

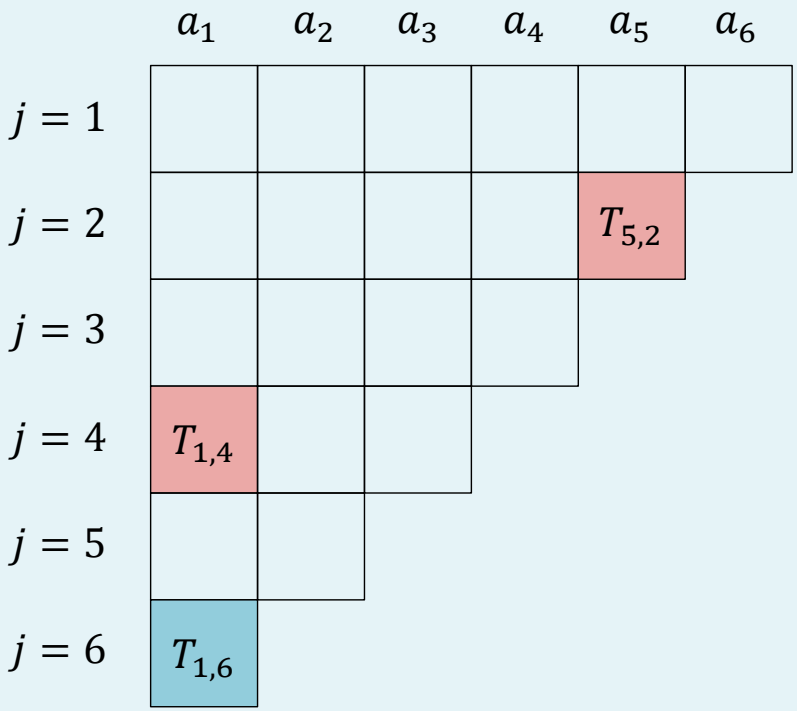


$x = a_1 a_2 a_3 a_4 a_5 \mid a_6$

$(A \longrightarrow BC) \in P,$

$B \in T_{1,5},\ C \in T_{6,1} \Rightarrow A \in T_{1,6}$

Algoritma CYK

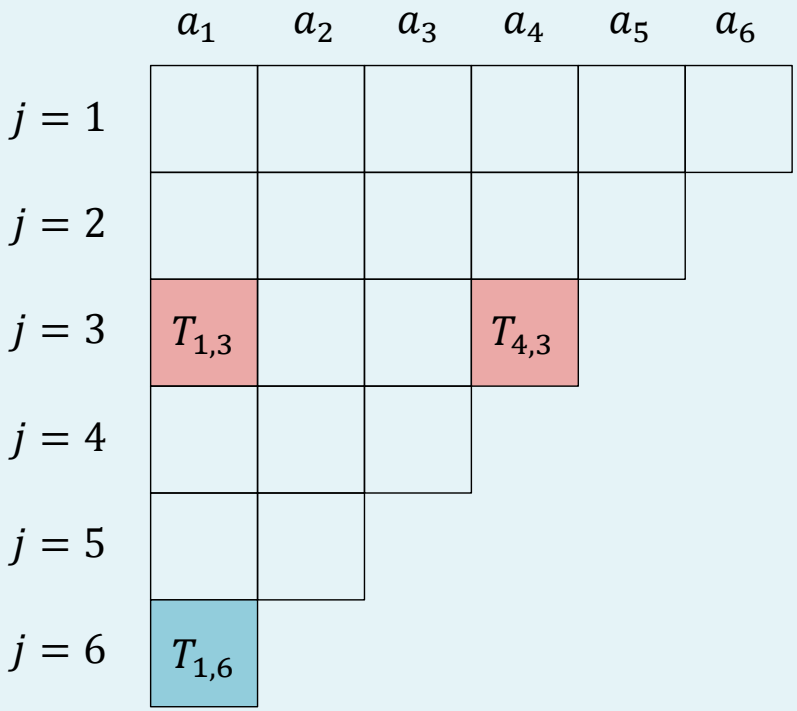


$x = a_1 a_2 a_3 a_4 \mid a_5 a_6$

$(A \longrightarrow BC) \in P,$

$B \in T_{1,4},\ C \in T_{5,2} \Rightarrow A \in T_{1,6}$

Algoritma CYK

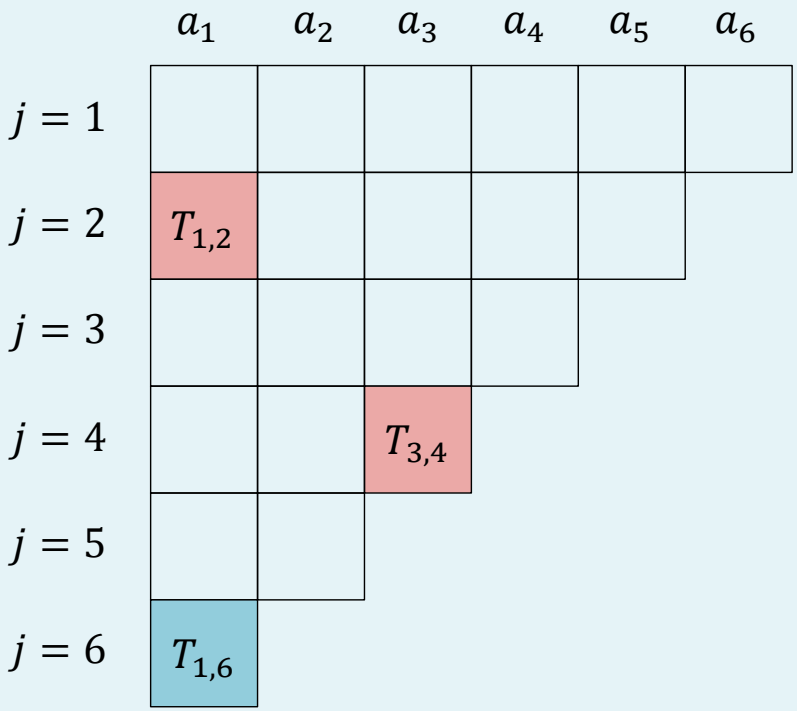


$$x = a_1 a_2 a_3 \mid a_4 a_5 a_6$$

$$(A \longrightarrow BC) \in P,$$

$$B \in T_{1,3},\ C \in T_{4,3} \Rightarrow A \in T_{1,6}$$

Algoritma CYK

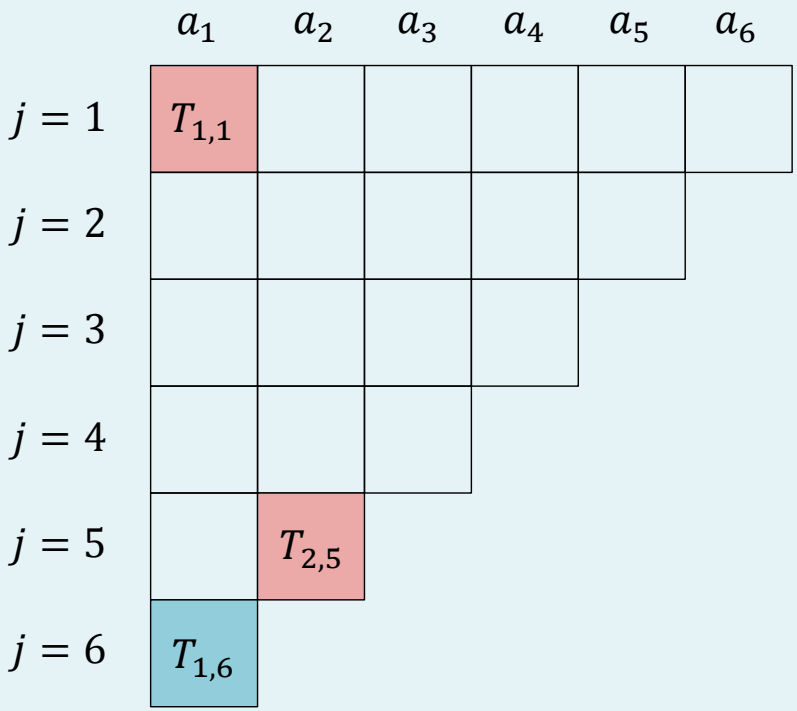


$x = a_1 a_2 \mid a_3 a_4 a_5 a_6$

$(A \longrightarrow BC) \in P,$

$B \in T_{1,2},\ C \in T_{3,4} \Rightarrow A \in T_{1,6}$

Algoritma CYK



$x = a_1 \mid a_2a_3a_4a_5a_6$

$(A \longrightarrow BC) \in P,$

$B \in T_{1,1},\ C \in T_{2,5} \Rightarrow A \in T_{1,6}$

Algoritma CYK

input $G = (V, \Sigma, P, S), w \in \Sigma^*$
$n := |w|$
for $i \in \{1, \ldots, n\}$ do
    $T_{i,1} := \{A \mid A \longrightarrow x_{i,1} \in P\}$
end
for $j \in \{2, \ldots, n\}$ do
    for $i \in \{1, \ldots, n - j + 1\}$ do
        $T_{i,j} := \emptyset$
        for $k \in \{1, \ldots, j - 1\}$ do
            $T_{i,j} := T_{i,j} \cup$
                $\{A \mid A \longrightarrow BC \in P$ dimana $B \in T_{i,k}, C \in T_{i+k,j-k}\}$
        end
    end
end
if $S \in T_{1,n}$ then
    return true
else
    return false

Algoritma CYK

Contoh 1: Diketahui grammar context free dengan production sebagai berikut:

$$S \longrightarrow AD \mid FG$$
$$D \longrightarrow SE \mid BC$$
$$E \longrightarrow BC$$
$$F \longrightarrow AF \mid a$$
$$G \longrightarrow BG \mid CG \mid b$$
$$H \longrightarrow SC$$
$$A \longrightarrow a$$
$$B \longrightarrow b$$
$$C \longrightarrow c$$

Pertanyaan: Diketahui $x = aabcbc$. Apakah word $x \in L$?

Algoritma CYK

Contoh 2: Diketahui grammar context free dengan production sebagai berikut:

$$S \longrightarrow AB$$
$$A \longrightarrow ab \mid aAb$$
$$B \longrightarrow c \mid cB$$

Pertanyaan: Diketahui $x = aaabbbcc$. Apakah word $x \in L$?

Algoritma CYK

Algoritma CYK adalah salah satu algoritma yang paling efisien yang dapat diterapkan pada grammar context free.

Pada prakteknya, algoritma ini akan lambat jika di parsing, contohnya pada bahasa pemrograman Java yang panjang.

Ada prosedur yang lebih efisien, namun hanya bisa digunakan pada sub-kelas bahasa context free. Pada prakteknya algoritma yang lebih sering digunakan adalah *recursive decent parser* dan *LR(k)-parser*.

Referensi

1. Hopcroft, Motwani, Ullman: *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, 2001
2. James A. Anderson: *Automata Theory with Modern Applications*, Cambridge University Press, 2006.
3. Uwe Schöning: *Theoretische Informatik – kurzgefaßt*. Spektrum, 2008. (5. Auflage)