Algorithm Design–II

(Dynamic Programming– **The CYK Algorithm**)

Presented
By

Dr. Rajesh Purkait
Asst. Professor
Dept. of CSE ITER
(S`O'A Deemed To Be University)
E-mail: rajeshpurkait@soa.ac.in

Outline

The CYK Algorithm

- *The membership problem*:
  - Problem:
    - Given a conteVt-free grammar **G** and a string **w**
      - **G** = (V, ∑ ,P , S) where
        - » V finite set of variables
        - » ∑ (the alphabet) finite set of terminal symbols
        - » P finite set of rules
        - » S start symbol (distinguished element of V)
        - » V and ∑ are assumed to be disjoint
      - **G** is used to generate the string of a language
  - Question:
    - Is **w** in **L**(**G**)?

The CYK Algorithm

- J. Cocke
- D. Younger,
- T. Kasami

  – Independently developed an algorithm to answer this question.

The CYK Algorithm Basics

- The Structure of the rules in a Chomsky Normal Form grammar
- Uses a "dynamic programming" or "table-filling algorithm"

Chomsky Normal Form

- *Normal Form* is described by a set of conditions that each rule in the grammar must satisfy
- ConteVt-free grammar is in CNF if each rule has one of the following forms:

  – A → BC at most 2 symbols on right side
  – A → a, or terminal symbol
  – S → λ null string
  where B, C ∈ V – {S}

Construct a Triangular Table

- Each row corresponds to one length of substrings
  - Bottom Row – Strings of length 1
  - Second from Bottom Row – Strings of length 2

    .

    .
  - Top Row – string ‘w’

Construct a Triangular Table

- $\mathbf{V_{i,\,i}}$ is the set of variables A such that $A \rightarrow w_i$ is a production of G

- Compare at most n pairs of previously computed sets:

  $(\mathbf{V_{i,\,i}}, \mathbf{V_{i+1,\,j}}), (\mathbf{V_{i,\,i+1}}, \mathbf{V_{i+2,\,j}}) \ldots (\mathbf{V_{i,\,j-1}}, \mathbf{V_{j,\,j}})$

Construct a Triangular Table

$V_{1,5}$				
$V_{1,4}$	$V_{2,4}$			
$V_{1,3}$	$V_{2,3}$	$V_{3,3}$		
$V_{1,2}$	$V_{2,2}$	$V_{3,2}$	$V_{4,2}$	
$V_{1,1}$	$V_{2,1}$	$V_{3,1}$	$V_{4,1}$	$V_{5,1}$
w_1	w_2	w_3	w_4	w_5

Table for string '**w**' that has length 5

Construct a Triangular Table

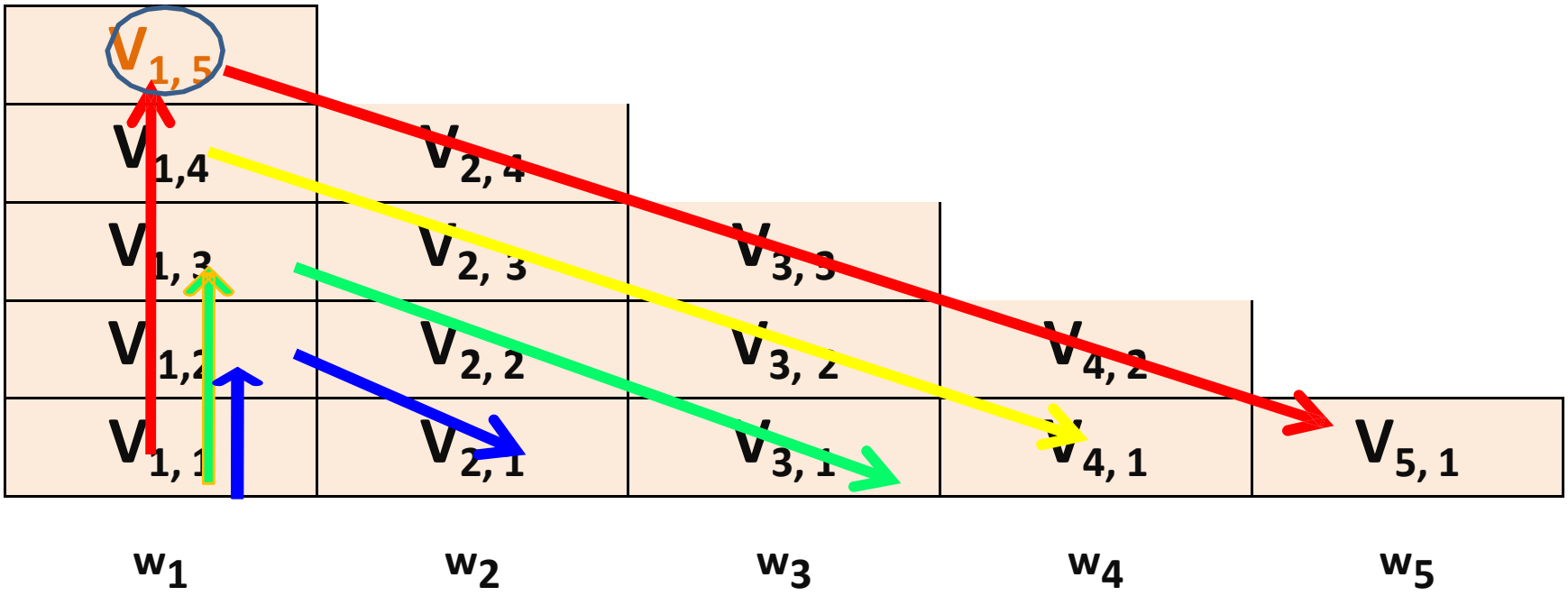

Table for string '**w**' that has length 5

EVample CYK Algorithm

- Show the CYK Algorithm with the following eVample:
  - CNF grammar **G**
    - S → AB | BC
    - A → BA | a
    - B → CC | b
    - C → AB | a
  - **w** is baaba
  - Question Is **baaba** in L(G)?

Constructing The Triangular Table

S → AB | BC
A → BA | a
B → CC | b
C → AB | a

{B}	**{A, C}**	**{A, C}**	**{B}**	**{A, C}**
b	**a**	**a**	**b**	**a**

Calculating the Bottom ROW

Constructing The Triangular Table

- $V_{1,2} = (V_{i,i}, V_{i+1,j}) = (V_{1,1}, V_{2,2})$
- ➔ {B}{A,C} = {BA, BC}
- Steps:
  - Look for production rules to generate BA or BC
  - There are two: S and A
  - $V_{1,2} = \{A, S\}$

S → AB | BC
A → BA | a
B → CC | b
C → AB | a

Constructing The Triangular Table

{A, S}				
{B}	{A, C}	{A, C}	{B}	{A, C}
b	a	a	b	a

Constructing The Triangular Table

- $V_{2,3} = (V_{i,i}, V_{i+1,j}) = (V_{2,2}, V_{3,3})$
- ➔ {A, C}{A,C} = {AA, AC, CA, CC} = Y
- Steps:
  - Look for production rules to generate Y
  - There is one: B
  - $V_{2,3} = \{B\}$

S → AB | BC
A → BA | a
B → CC | b
C → AB | a

Constructing The Triangular Table

{S, A}	{B}			
{B}	{A, C}	{A, C}	{B}	{A, C}
b	a	a	b	a

Constructing The Triangular Table

- $V_{3,4} = (V_{i,i}, V_{i+1,j}) = (V_{3,3}, V_{4,4})$
- ➔ {A, C}{B} = {AB, CB} = Y
- Steps:
  - Look for production rules to generate Y
  - There are two: S and C
  - $V_{3,4} = \{S, C\}$

S → AB | BC
A → BA | a
B → CC | b
C → AB | a

Constructing The Triangular Table

{A, S}	{B}	{S, C}		
{B}	{A, C}	{A, C}	{B}	{A, C}
b	a	a	b	a

Constructing The Triangular Table

- $V_{4,5} = (V_{i,i}, V_{i+1,j}) = (V_{4,4}, V_{5,5})$
- ➔ {B}{A, C} = {BA, BC} = Y
- Steps:
  - Look for production rules to generate Y
  - There are two: S and A
  - $V_{4,5} = \{A, S\}$

S → AB | BC
A → BA | a
B → CC | b
C → AB | a

Constructing The Triangular Table

{A, S}	{B}	{S, C}	{A, S}	
{B}	{A, C}	{A, C}	{B}	{A, C}
b	a	a	b	a

Constructing The Triangular Table

- $V_{1,3} = (V_{i,i}, V_{i+1,j})\,(V_{i,i+1}, V_{i+2,j})$
  $= (V_{1,1}, V_{2,3}), (V_{1,2}, V_{3,3})$
- ➔ {B}{B} **U** {A,S}{A, C}= {BB, AA, AC, SA, SC} = Y
- Steps:
  - Look for production rules to generate Y
  - There are NONE: S and A
  - $V_{1,3} = \emptyset$
  - **no elements in this set (empty set)**

S → AB | BC
A → BA | a
B → CC | b
C → AB | a

Constructing The Triangular Table

Ø				
{S, A}	{B}	{S, C}	{S, A}	
{B}	{A, C}	{A, C}	{B}	{A, C}
b	a	a	b	a

Constructing The Triangular Table

- $V_{2,4} = (V_{i,i}, V_{i+1,j})\,(V_{i,i+1}, V_{i+2,j})$
  $= (V_{2,2}, V_{3,4}), (V_{2,3}, V_{4,4})$
- ➔ {A, C}{S, C} **U** {B}{B}= {AS, AC, CS, CC, BB} = Y
- Steps:
  - Look for production rules to generate Y
  - There is one: B
  - $V_{2,4} = \{B\}$

S → AB | BC
A → BA | a
B → CC | b
C → AB | a

Constructing The Triangular Table

Ø	{B}			
{ A,S}	{B}	{S, C}	{ A, S}	
{B}	{A, C}	{A, C}	{B}	{A, C}
b	a	a	b	a

Constructing The Triangular Table

- $V_{3,5} = (V_{i,i}, V_{i+1,j})\,(V_{i,i+1}, V_{i+2,j})$

  $= (V_{3,3}, V_{4,5}), (V_{3,4}, V_{5,5})$

- ➔ {A,C}{A,S} **U** {S,C}{A,C}

  = {AA, AS, CS, CA, SA, SC, CA, CC} = Y

- Steps:
  - Look for production rules to generate Y
  - There is one: B
  - $V_{3,5} = \{B\}$

S → AB | BC
A → BA | a
B → CC | b
C → AB | a

Constructing The Triangular Table

Ø	{B}	{B}		
{A, S}	{B}	{S, C}	{A, S}	
{B}	{A, C}	{A, C}	{B}	{A, C}
b	a	a	b	a

Final Triangular Table

{A, S, C}	← $V_{1,5}$			
Ø	{A, S, C}			
Ø	{B}	{B}		
{A, S}	{B}	{S, C}	{A, S}	
{B}	{A, C}	{A, C}	{B}	{A, C}
b	a	a	b	a

- Table for string '**w**' that has length 5
- The algorithm populates the triangular table

EVample (Result)

- Is baaba in L(G)?

Yes

We can see the S in the set V_{1n} where 'n' = 5 We can see the table the cell V_{15} = {A, S, C} then

if S Є V_{15} then baaba Є L(G)

Theorem

- The CYK Algorithm correctly computes V_{ij} for all i and j; thus w is in L(G) if and only if S is in V_{1n}.
- The running time of the algorithm is $O(n^3)$.

References

- J. E. Hopcroft, R. Motwani, J. D. Ullman, *Introduction to Automata Theory, Languages and Computation*, Second Edition, Addison Wesley, 2001
- T.A. Sudkamp, An Introduction to the Theory of Computer Science Languages and Machines, Third Edition, Addison Wesley, 2006

Question

- Show the CYK Algorithm with the following eVample:
  - CNF grammar **G**
    - S → AB | BC
    - A → BA | a
    - B → CC | b
    - C → AB | a
  - **w** is ababa
  - Question Is **ababa** in L(G)?
- Basics of CYK Algorithm
  - The Structure of the rules in a Chomsky Normal Form grammar
  - Uses a “dynamic programming” or “table-filling algorithm”
- CompleVity O(n3)

Thank You