The CYK Algorithm

- *The membership problem*:
  - Problem:
    - Given a context-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
- Context-free grammar is in CNF if each rule has one of the following forms:
  - $A \rightarrow BC$ at most 2 symbols on right side
  - $A \rightarrow a$, or terminal symbol
  - $S \rightarrow \lambda$ null string

  where $B, C \in V - \{S\}$

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

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

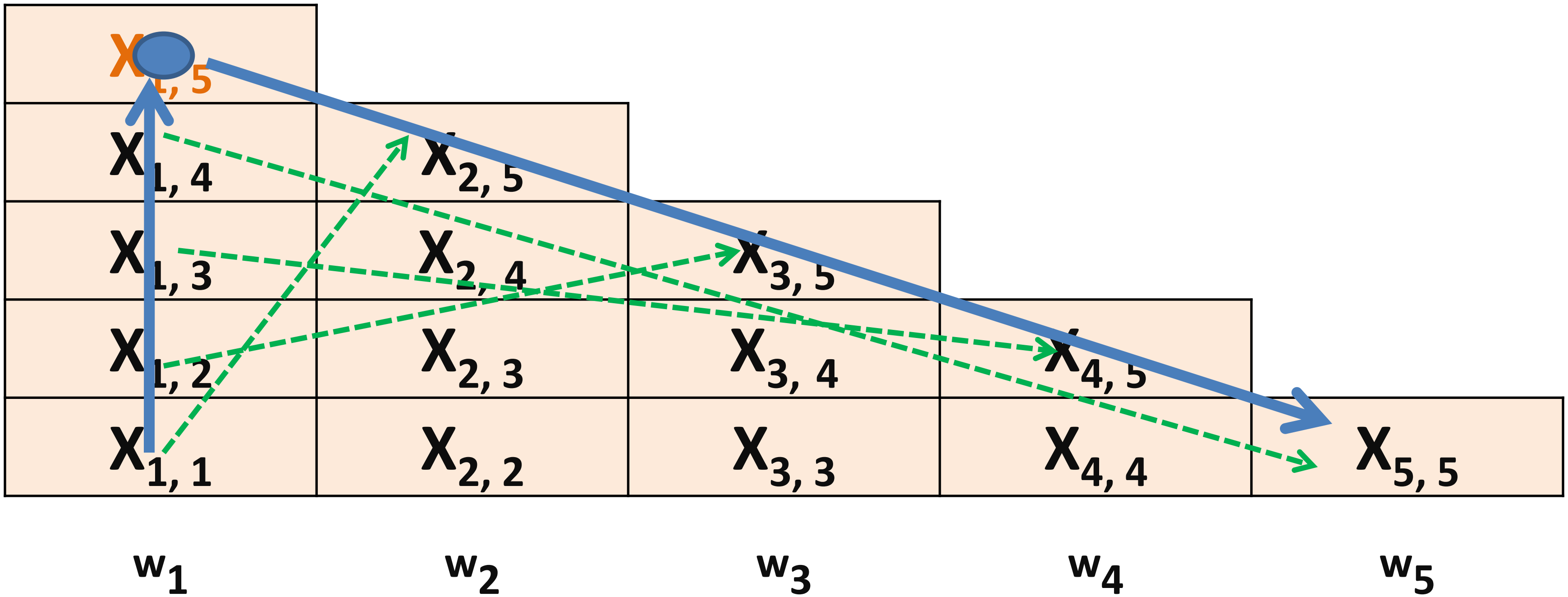
- Compare at most n pairs of previously computed sets:
  $(\mathbf{X_{i,\,i}}\,\mathbf{,}\;\mathbf{X_{i+1,\,j}}),\ (\mathbf{X_{i,\,i+1}}\,\mathbf{,}\;\mathbf{X_{i+2,\,j}}) \ldots (\mathbf{X_{i,\,j-1}}\,\mathbf{,}\;\mathbf{X_{j,\,j}}\,\mathbf{)}$

Construct a Triangular Table

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

Table for string ‘**w**’ that has length 5

Construct a Triangular Table



Looking for pairs to compare

Example CYK Algorithm

- Show the CYK Algorithm with the following example:
  - 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

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

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

Constructing The Triangular Table

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

Constructing The Triangular Table

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

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

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

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

Constructing The Triangular Table

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

Constructing The Triangular Table

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

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

Constructing The Triangular Table

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

Constructing The Triangular Table

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

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

Constructing The Triangular Table

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

Constructing The Triangular Table

- $\mathbf{X_{2,4} = (X_{i,i}, X_{i+1,j})\ (X_{i,i+1}, X_{i+2,j})}$

  $\mathbf{= (X_{2,2}, X_{3,4}), (X_{2,3}, X_{4,4})}$
- ➔ {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
  - $\mathbf{X_{2,4} = \{B\}}$

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

Constructing The Triangular Table

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

Constructing The Triangular Table

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

  $\mathbf{= (X_{3,3}, X_{4,5}), (X_{3,4}, X_{5,5})}$
- ➔ {A,C}{S,A} **U** {S,C}{A,C}

  = {AS, AA, CS, CA, SA, SC, CA, CC} = Y
- Steps:
  - Look for production rules to generate Y
  - There is one: B
  - $\mathbf{X_{3,5} = \{B\}}$

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

Constructing The Triangular Table

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

Final Triangular Table

{S, A, C}	$\leftarrow X_{1, 5}$			
Ø	{S, A, C}			
Ø	{B}	{B}		
{S, A}	{B}	{S, C}	{S, A}	
{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

Example (Result)

- Is baaba in L(G)?

Yes

We can see the S in the set X_{1n} where ‘n’ = 5
We can see the table
the cell X_{15} = (S, A, C) then
if S $\in$ X_{15} then baaba $\in$ L(G)

Theorem

- The CYK Algorithm correctly computes $X_{i\,j}$ for all i and j; thus w is in L(G) if and only if S is in X_{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 example:
  - CNF grammar **G**
    - $S \rightarrow AB \mid BC$
    - $A \rightarrow BA \mid a$
    - $B \rightarrow CC \mid b$
    - $C \rightarrow AB \mid a$
  - **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”
- Complexity O(n3)