CKY Parser

Instructor : Dr. Hanaa Bayomi Ali
Mail : h.mobarz @ fci-cu.edu.eg

Parsing Algorithms

Top-down vs. bottom-up:

- Top-down: (goal-driven): from the start symbol down.
- Bottom-up: (data-driven): from the symbols up.

Naive vs. dynamic programming:

- Naive: enumerate everything.
- Backtracking: try something, discard partial solutions.
- Dynamic programming: save partial solutions in a table.

Examples:

- CKY: bottom-up dynamic programming.
- Early parsing: top-down dynamic programming.

CKY (Cocke-Kasami-Younger)

- One of the earliest recognition and parsing algorithms
- The standard version of CKY can only recognize languages defined by context-free grammars in Chomsky Normal Form (CNF).
- It is also possible to extend the CKY algorithm to handle some grammars which are not in CNF
  - Harder to understand
- Based on a “dynamic programming” approach:
  - Build solutions compositionally from sub-solutions
- Uses the grammar directly.

Chomsky Normal Form (CNF)

A context-free grammar where the right side of each production rule is restricted to be either two non terminals or one terminal. Production can be one of following formats:

- $A \rightarrow \alpha$
- $A \rightarrow BC$

Any CFG can be converted to a weakly equivalent grammar in CNF

CNF Conversion

Three main conditions:

1) Hybrid rules:

INF-VP $\rightarrow$ to VP

2) Unit productions:

A $\rightarrow$ B

3) Long productions:

A $\rightarrow$ B C D

CNF Conversion

1) Hybrid rule conversion:

- Replace all terminals with dummy non-terminals
- E.g. INF-VP → to VP
  - To → to , INF-VP → To VP

2) Unit productions:

- Rewrite RHS with RHS of all derivable non-unit productions
- If A⇒ B and B → w , then add A → w
-

3) Long productions:

- Introduce new non-terminals and spread over rules
- A → B C D
  - A →ED , E → BC

CKY Algorithm for Deciding CFL

Consider the grammar G given by:

S → A B

S → X B

T → A B

T → X B

X→ A T

A → a

B → b

Is w = aaabbb in L(G) ?

CKY Algorithm for Deciding CFL

Now look at aaabbb :

a	*a*	*a*	*b*	*b*	*b*

S → A B

S → X B

T → A B

T → X B

X → A T

A → a

B → b

CKY Algorithm for Deciding CFL

1) Write variables for all length 1 substrings.

S → A B

S → X B

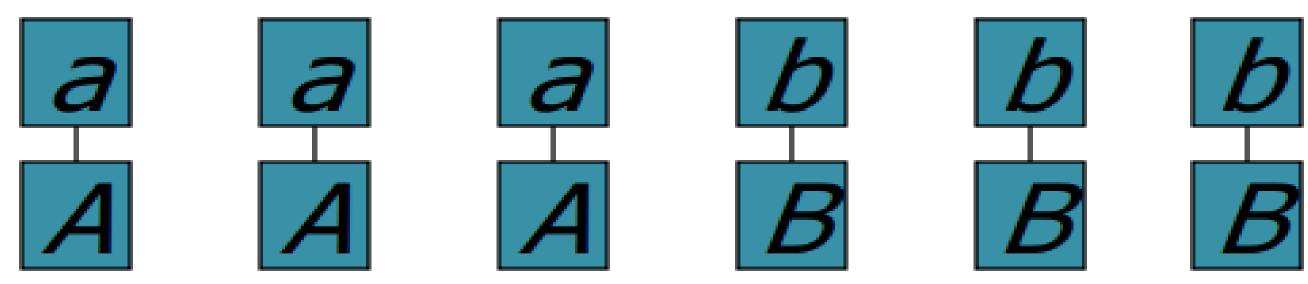
T → A B

T → X B

X → A T

A → a

B → b



CKY Algorithm for Deciding CFL

2) Write variables for all length 2 substrings.

S → A B

S → X B

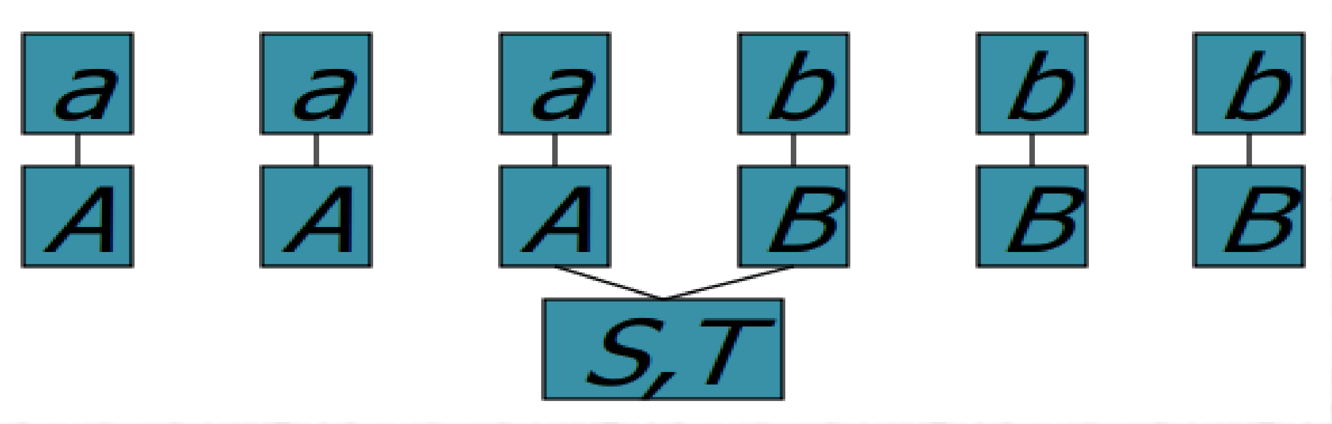
T → A B

T → X B

X → A T

A → a

B → b



CKY Algorithm for Deciding CFL

3) Write variables for all length 3 substrings.

S → A B

S → X B

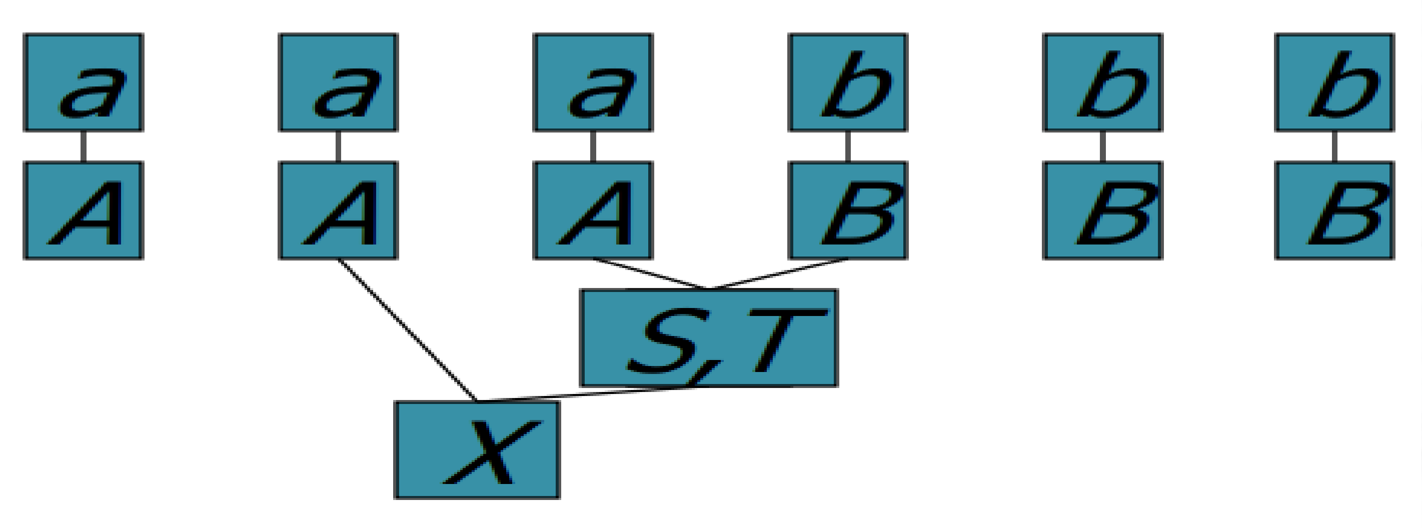
T → A B

T → X B

X→ A T

A → a

B → b



CKY Algorithm for Deciding CFL

3) Write variables for all length 4 substrings.

S → A B

S → X B

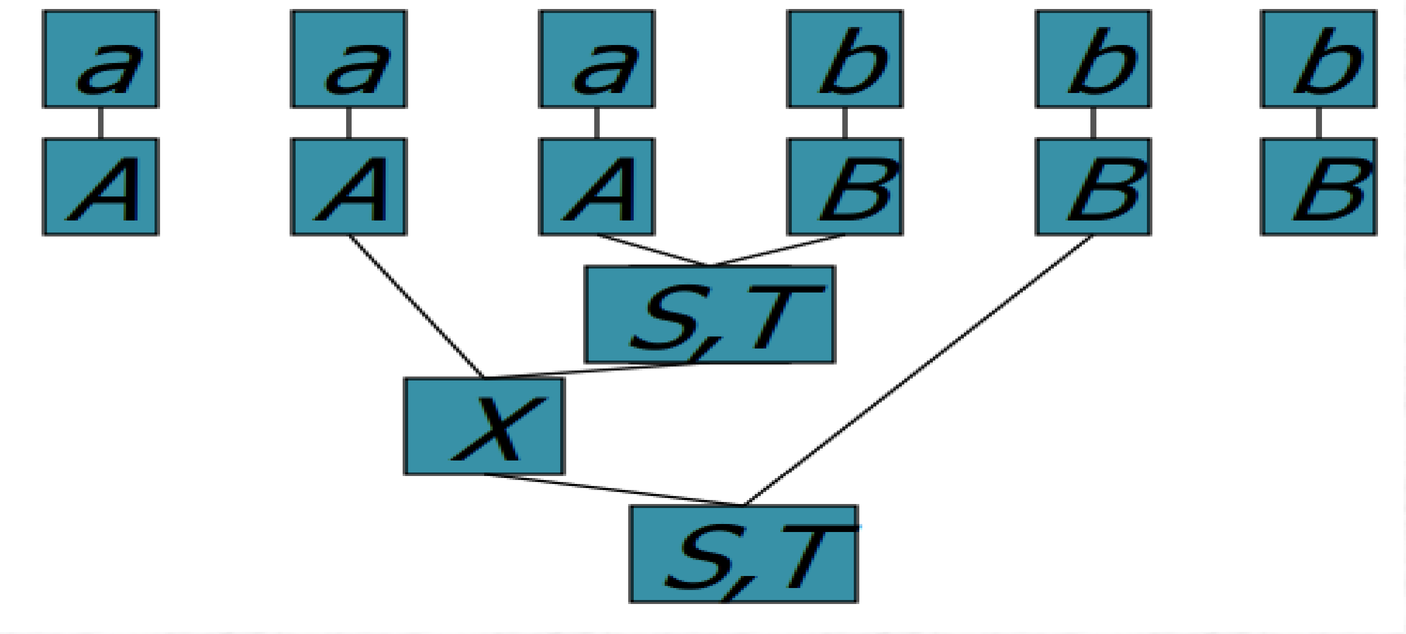
T → A B

T → X B

X→ A T

A → a

B → b



CKY Algorithm for Deciding CFL

3) Write variables for all length 5 substrings.

S → A B

S → X B

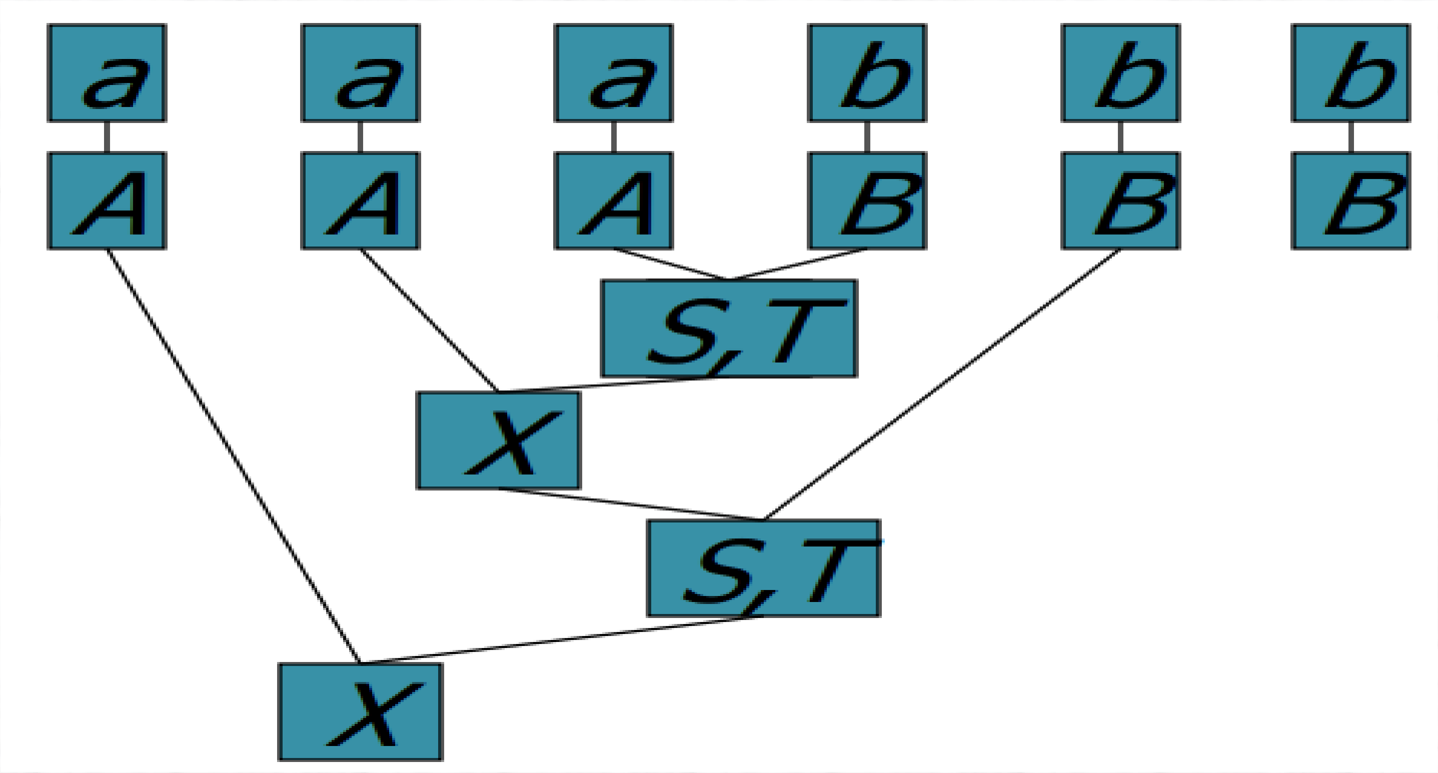
T → A B

T → X B

X → A T

A → a

B → b



CKY Algorithm for Deciding CFL

3) Write variables for all length 6 substrings.

S → A B

S → X B

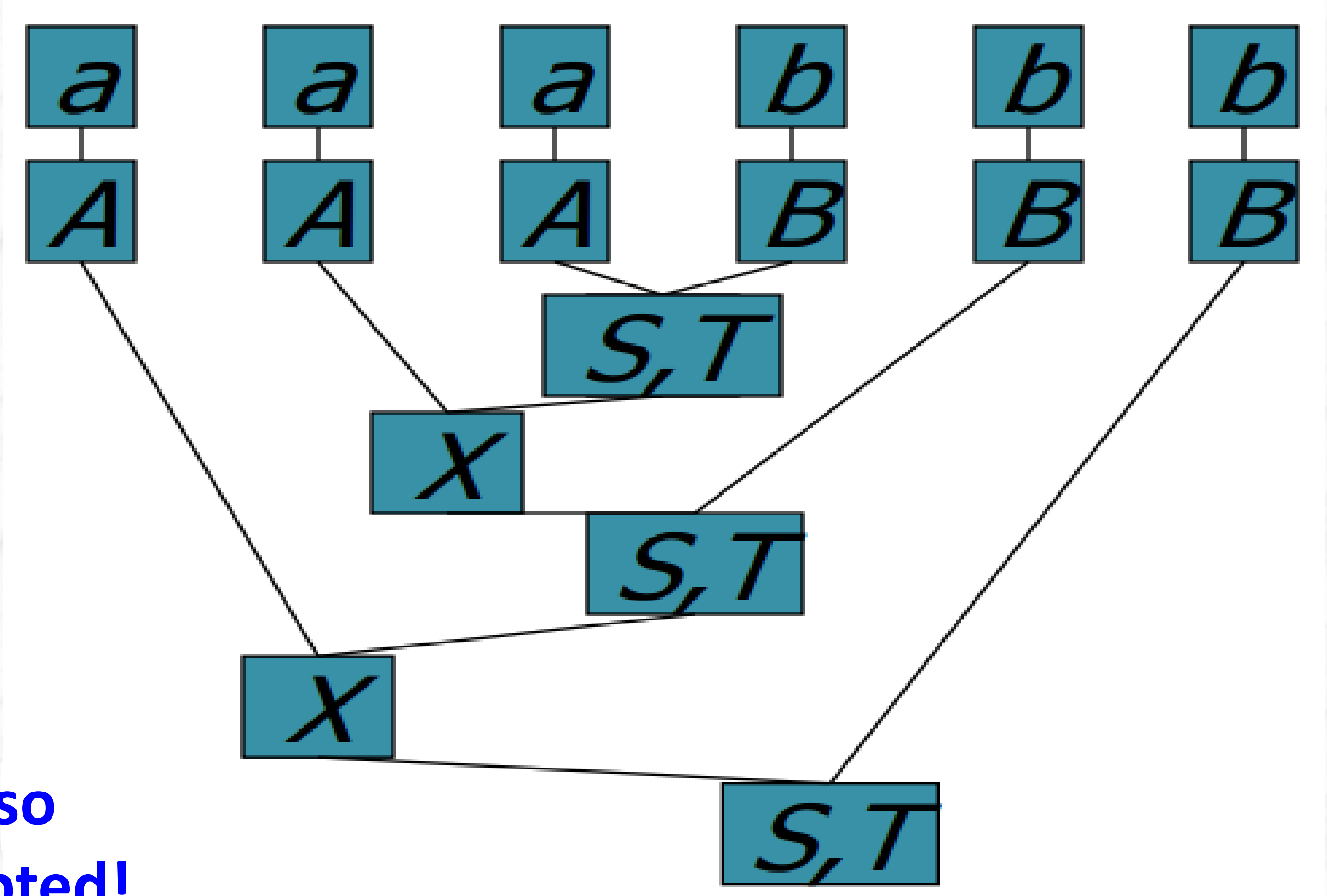
T → A B

T → X B

X→ A T

A → a

B → b



S is included so aaabbb accepted!

The CKY Algorithm

function CKY (word w, grammar P) returns table

for i $\leftarrow$ from 1 to LENGTH(w) do

table[i-1, i] $\leftarrow$ $\{A \mid A \rightarrow w_i \in P\}$

for j $\leftarrow$ from 2 to LENGTH(w) do

for i $\leftarrow$ from j-2 down to 0 do

for k $\leftarrow$ i + 1 to j – 1 do

table[i,j] $\leftarrow$ table[i,j] $\cup$ $\{A \mid A \rightarrow BC \in P,$

$B \in$ table[i,k], $C \in$ table[k,j] $\}$

If the start symbol $S \in$ table[0,n] then $w \in L(G)$

CKY Algorithm for Deciding CFL

The table chart used by the algorithm:

i \ j	1 a	2 a	3 a	4 b	5 b	6 b
0						
1						
2						
3						
4						
5						

Build an n+1 x n+1 matrix, where n = number of words in input

CKY Algorithm for Deciding CFL

The table chart used by the algorithm:

i \ j	1 a	2 a	3 a	4 b	5 b	6 b
0	[0,1]					
1		[1,2]				
2			[2,3]			
3				[3,4]		
4					[4,5]	
5						[5,6]

Illustrate the numbering of cells: [i,j]'s represent spans

CKY Algorithm for Deciding CFL

The table chart used by the algorithm:

i \ j	1 a	2 a	3 a	4 b	5 b	6 b
0	[0,1] A					
1		[1,2] A				
2			[2,3]			
3				[3,4]		
4					[4,5]	
5						[5,6]

S → A B
S → X B
T → A B
T → X B
X→ A T
A → a
B → b

AA get nothing

CKY Algorithm for Deciding CFL

The table chart used by the algorithm:

i \ j	1 a	2 a	3 a	4 b	5 b	6 b
0	[0,1] A					
1		[1,2] A				
2			[2,3] A			
3				[3,4]		
4					[4,5]	
5						[5,6]

S → A B
S → X B
T → A B
T → X B
X → A T
A → a
B → b

'a' is labeled A

CKY Algorithm for Deciding CFL

The table chart used by the algorithm:

i \ j	1 a	2 a	3 a	4 b	5 b	6 b
0	[0,1] A					
1		[1,2] A				
2			[2,3] A			
3				[3,4] B		
4					[4,5]	
5						[5,6]

S → A B
S → X B
T → A B
T → X B
X → A T
A → a
B → b

‘b’ is labeled B

CKY Algorithm for Deciding CFL

The table chart used by the algorithm:

i \ j	1 a	2 a	3 a	4 b	5 b	6 b
0	[0,1] A					
1		[1,2] A				
2			[2,3] A			
3				[3,4] B		
4					[4,5]	
5						[5,6]

S → A B
S → X B
T → A B
T → X B
X → A T
A → a
B → b

Found S and T [2,3],[3,4] **S ->AB T->AB**

CKY Algorithm for Deciding CFL

The table chart used by the algorithm:

i \ j	1 a	2 a	3 a	4 b	5 b	6 b
0	[0,1] A					
1		[1,2] A				
2			[2,3] A	[2,4] S,T		
3				[3,4] B		
4					[4,5]	
5						[5,6]

S → A B
S → X B
T → A B
T → X B
X→ A T
A → a
B → b

Found S and T [2,3],[3,4] **S ->AB T->AB**

CKY Algorithm for Deciding CFL

The table chart used by the algorithm:

i \ j	1 a	2 a	3 a	4 b	5 b	6 b
0	[0,1] A					
1		[1,2] A		[1,4] X		
2			[2,3] A	[2,4] S,T		
3				[3,4] B		
4					[4,5]	
5						[5,6]

- S → A B
- S → X B
- T → A B
- T → X B
- X → A T
- A → a
- B → b

Test all filled cell with column = the row of current cell   Failed (AS)

CKY Algorithm for Deciding CFL

The table chart used by the algorithm:

i \ j	1 a	2 a	3 a	4 b	5 b	6 b
0	[0,1] A					
1		[1,2] A		[1,4] X		
2			[2,3] A	[2,4] S,T		
3				[3,4] B		
4					[4,5]	
5						[5,6]

Test all filled cell with column = the row of current cell Failed (AX)

CKY Algorithm for Deciding CFL

The table chart used by the algorithm:

i \ j	1 a	2 a	3 a	4 b	5 b	6 b
0	[0,1] A					
1		[1,2] A		[1,4] X		
2			[2,3] A	[2,4] S,T		
3				[3,4] B		
4					[4,5] B	
5						[5,6]

- S → A B
- S → X B
- T → A B
- T → X B
- X → A T
- A → a
- B → b

'b' is labeled B

CKY Algorithm for Deciding CFL

The table chart used by the algorithm:

i \ j	1 a	2 a	3 a	4 b	5 b	6 b
0	[0,1] A					
1		[1,2] A		[1,4] X		
2			[2,3] A	[2,4] S,T		
3				[3,4] B		
4					[4,5] B	
5						[5,6]

S → A B
S → X B
T → A B
T → X B
X → A T
A → a
B → b

‘BB not found

CKY Algorithm for Deciding CFL

The table chart used by the algorithm:

i \ j	1 a	2 a	3 a	4 b	5 b	6 b
0	[0,1] A					
1		[1,2] A		[1,4] X		
2			[2,3] A	[2,4] S,T		
3				[3,4] B		
4					[4,5] B	
5						[5,6]

S → A B
S → X B
T → A B
T → X B
X → A T
A → a
B → b

S B or TB not found

CKY Algorithm for Deciding CFL

The table chart used by the algorithm:

i \ j	1 a	2 a	3 a	4 b	5 b	6 b
0	[0,1] A					
1		[1,2] A		[1,4] X	[1,5] T,S	
2			[2,3] A	[2,4] S,T		
3				[3,4] B		
4					[4,5] B	
5						[5,6]

S → A B
S → X B
T → A B
T → X B
X → A T
A → a
B → b

'XB' produce T from [1,4] ,[4,5]

CKY Algorithm for Deciding CFL

The table chart used by the algorithm:

i \ j	1 a	2 a	3 a	4 b	5 b	6 b
0	[0,1] A				[0,5] X	
1		[1,2] A		[1,4] X	[1,5] T,S	
2			[2,3] A	[2,4] S,T		
3				[3,4] B		
4					[4,5] B	
5						[5,6]

S → A B
S → X B
T → A B
T → X B
X → A T
A → a
B → b

‘AT’ produce X from [0,1], [1,5]

CKY Algorithm for Deciding CFL

The table chart used by the algorithm:

i \ j	1 a	2 a	3 a	4 b	5 b	6 b
0	[0,1] A				[0,5] X	
1		[1,2] A		[1,4] X	[1,5] T,S	
2			[2,3] A	[2,4] S,T		
3				[3,4] B		
4					[4,5] B	
5						[5,6] B

$S \rightarrow A\ B$
$S \rightarrow X\ B$
$T \rightarrow A\ B$
$T \rightarrow X\ B$
$X \rightarrow A\ T$
$A \rightarrow a$
$B \rightarrow b$

'b' is labeled B

CKY Algorithm for Deciding CFL

The table chart used by the algorithm:

i \ j	1 a	2 a	3 a	4 b	5 b	6 b
0	[0,1] A				[0,5] X	
1		[1,2] A		[1,4] X	[1,5] T,S	
2			[2,3] A	[2,4] S,T		
3				[3,4] B		
4					[4,5] B	
5						[5,6] B

S → A B
S → X B
T → A B
T → X B
X→ A T
A → a
B → b

BB not found

CKY Algorithm for Deciding CFL

The table chart used by the algorithm:

i \ j	1 a	2 a	3 a	4 b	5 b	6 b
0	[0,1] A				[0,5] X	
1		[1,2] A		[1,4] X	[1,5] T,S	
2			[2,3] A	[2,4] S,T		
3				[3,4] B		
4					[4,5] B	
5						[5,6] B

S → A B
S → X B
T → A B
T → X B
X→ A T
A → a
B → b

'**TB**' not found

CKY Algorithm for Deciding CFL

The table chart used by the algorithm:

i \ j	1 a	2 a	3 a	4 b	5 b	6 b
0	[0,1] A				[0,5] X	[0,6] S,T
1		[1,2] A		[1,4] X	[1,5] T,S	
2			[2,3] A	[2,4] S,T		
3				[3,4] B		
4					[4,5] B	
5						[5,6] B

S → A B

S → X B

T → A B

T → X B

X → A T

A → a

B → b

XB produce S,T from [0,5],[5,6]

CKY Algorithm for Deciding CFL

The table chart used by the algorithm:

i \ j	1 a	2 a	3 a	4 b	5 b	6 b
0	[0,1] A				[0,5] X	[0,6] S,T
1		[1,2] A		[1,4] X	[1,5] T,S	
2			[2,3] A	[2,4] S,T		
3				[3,4] B		
4					[4,5] B	
5						[5,6] B

Found S node: [0,5] [5,6] Recognition algorithm returns True when a root node is found in [0,n]

Parsing results

- We keep the results for every **wij** in a table.

- Note that we only need to fill in entries up to the diagonal.

- Every entry in the table T[i,j] can contains up to r=|N| symbols (the size of the non-terminal set).

- We then want to find T[0,n,S] = true.