The CYK Parser

The CYK Membership Algorithm

Input:

- Grammar G in Chomsky Normal Form
- String w

Output:

find if $w \in L(G)$

The Algorithm

Input example:

- Grammar G:

$$S \to AB$$

$$A \to BB$$

$$A \to a$$

$$B \to AB$$

$$B \to b$$

- String w : $aabbb$

aabbb

a a b b b

aa ab bb bb

aab abb bbb

aabb abbb

aabbb

$S \rightarrow AB$

$A \rightarrow BB$

$A \rightarrow a$

$B \rightarrow AB$

$B \rightarrow b$

a	a	b	b	b
A	A	B	B	B
aa	ab	bb	bb	
aab	abb	bbb		
aabb	abbb			
aabbb				

$S \rightarrow AB$

$A \rightarrow BB$

$A \rightarrow a$

$B \rightarrow AB$

$B \rightarrow b$

a	a	b	b	b
A	A	B	B	B
aa	ab	bb	bb	
	S,B	A	A	
aab	abb	bbb		
aabb	abbb			
aabbb				

$S \rightarrow AB$

$A \rightarrow BB$

$A \rightarrow a$

$B \rightarrow AB$

$B \rightarrow b$

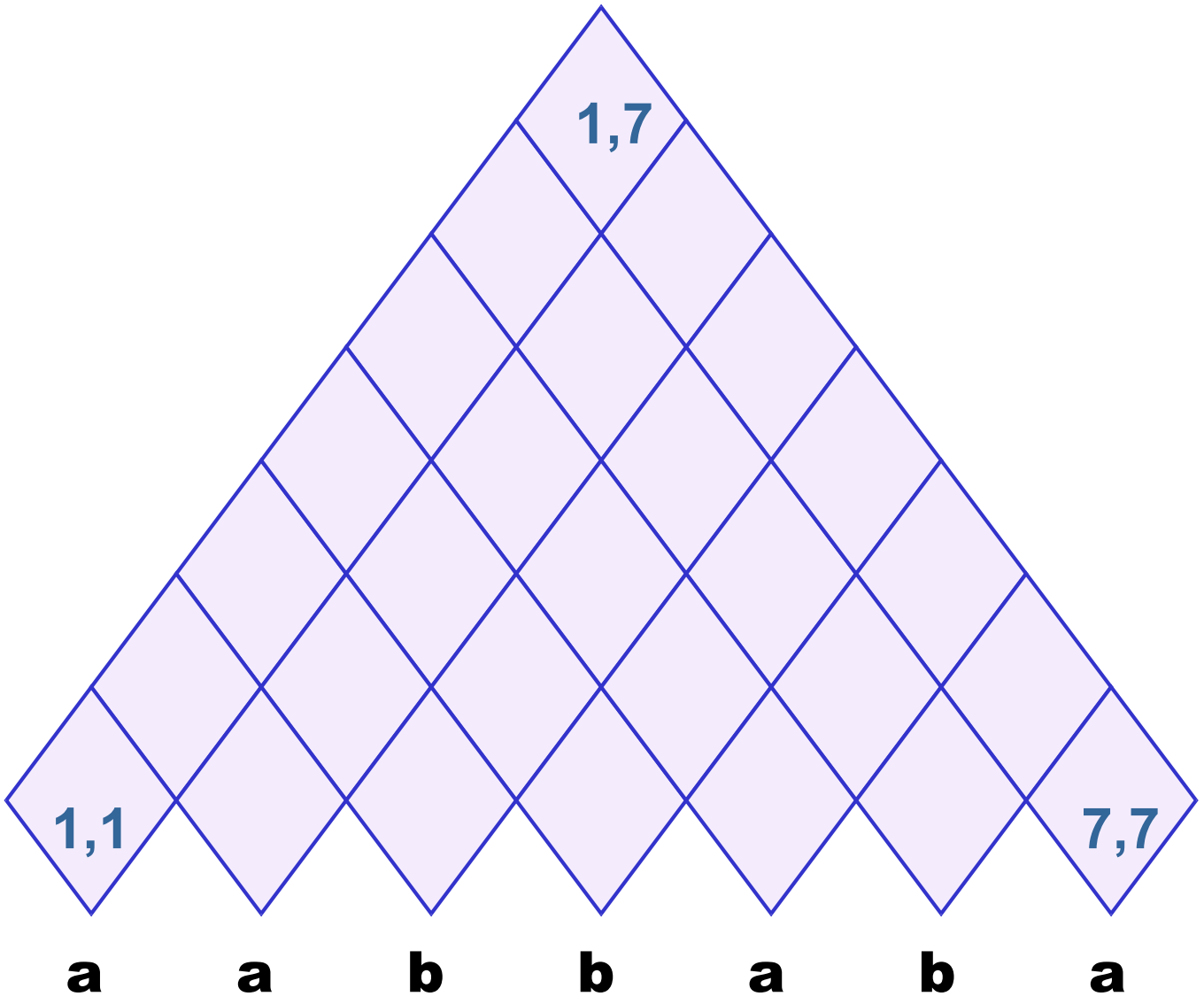
a	a	b	b	b
A	A	B	B	B
aa	ab	bb	bb	
	S,B	A	A	
aab	abb	bbb		
S,B	A	S,B		
aabb	abbb			
A	S,B			
aabbb				
S,B				

Therefore: $aabbb \in L(G)$

Time Complexity: $|w|^3$

Observation: The CYK algorithm can be easily converted to a parser (bottom up parser)

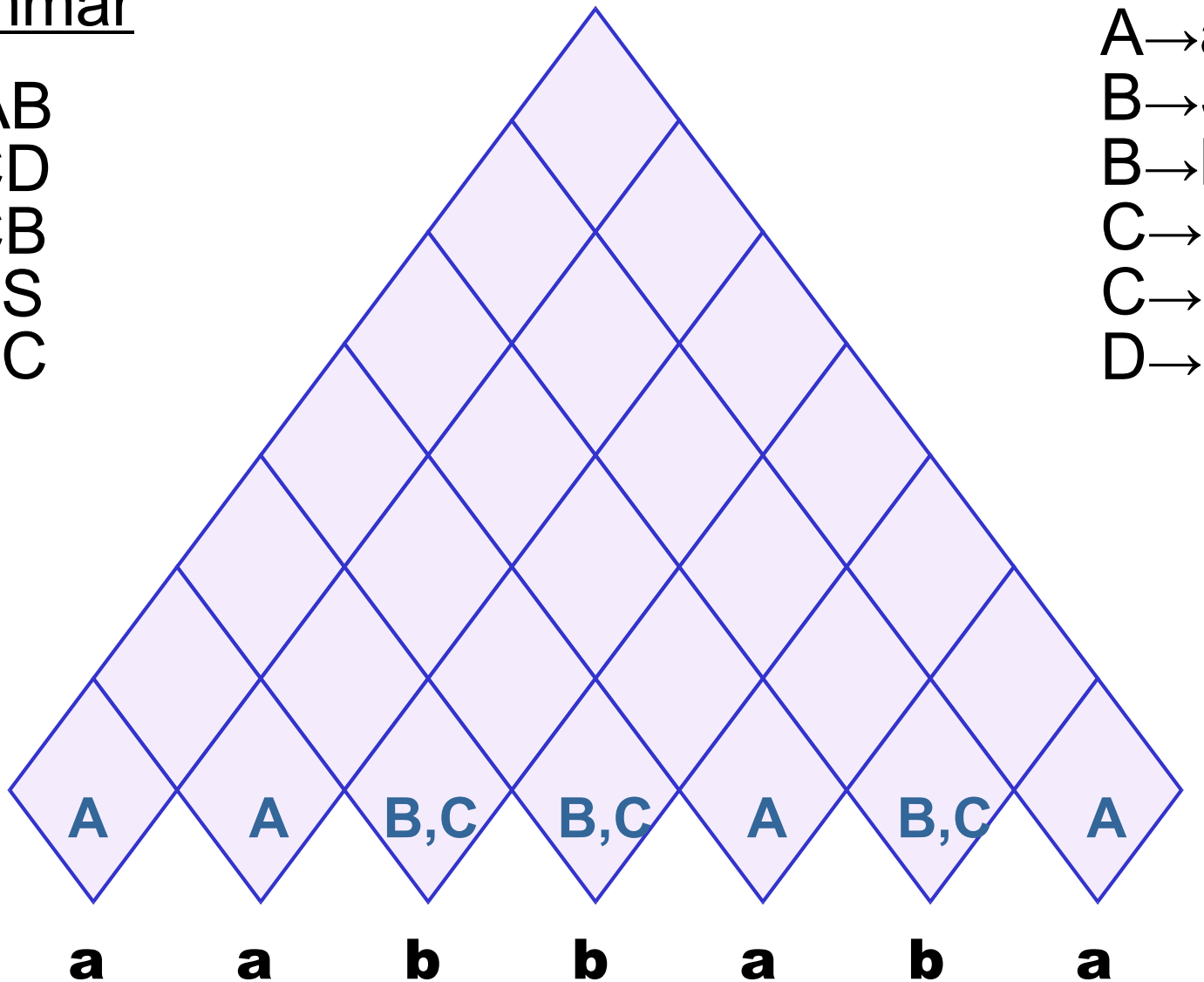
The following slides are courtesy of
Professor Papp, University of Debrecen.


1,7
1,1
7,7
a
a
b
b
a
b
a

Grammar

S→AB
S→CD
S→CB
S→SS
A→BC

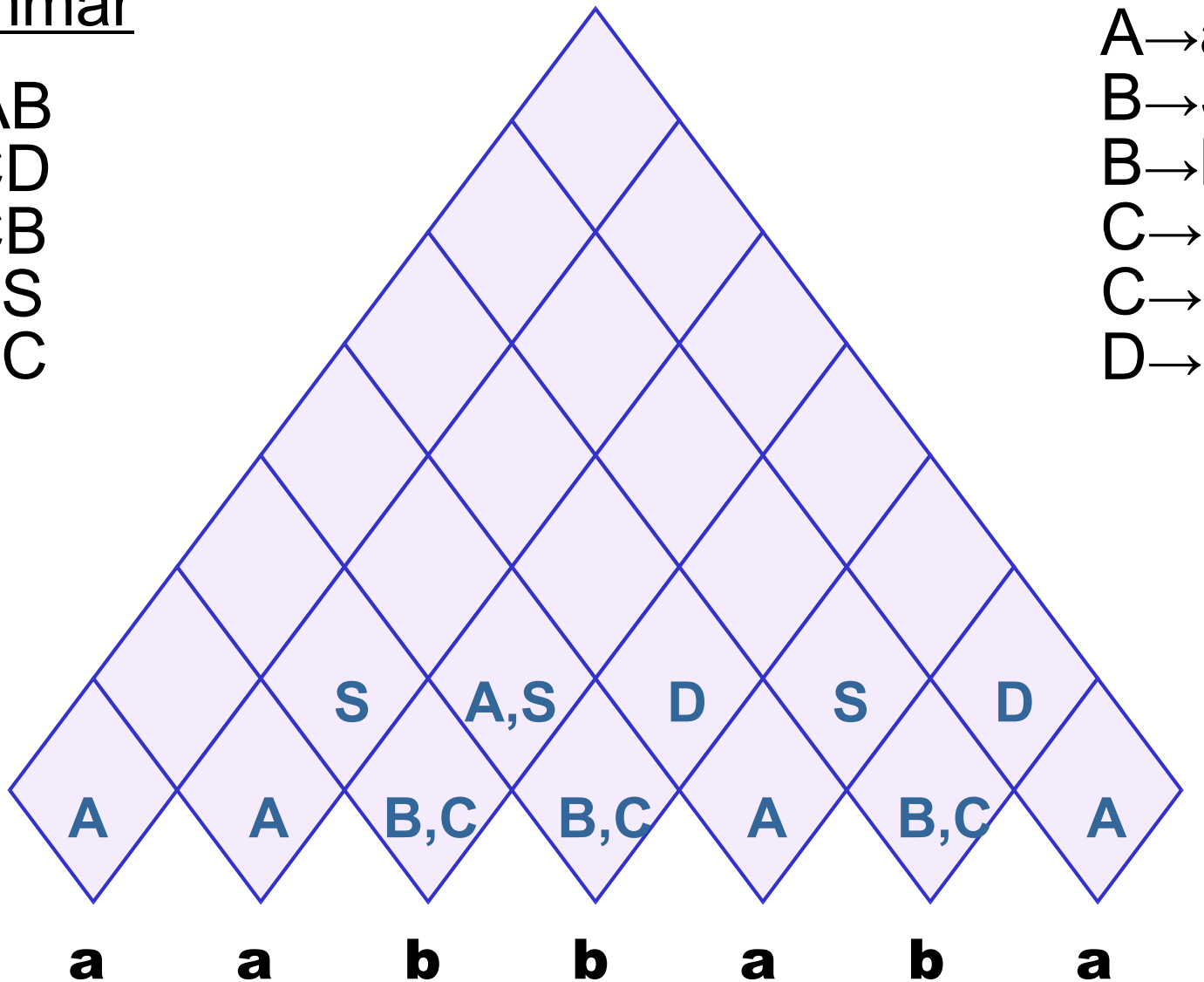
A→a
B→SC
B→b
C→DD
C→b
D→BA



Grammar

S→AB
S→CD
S→CB
S→SS
A→BC

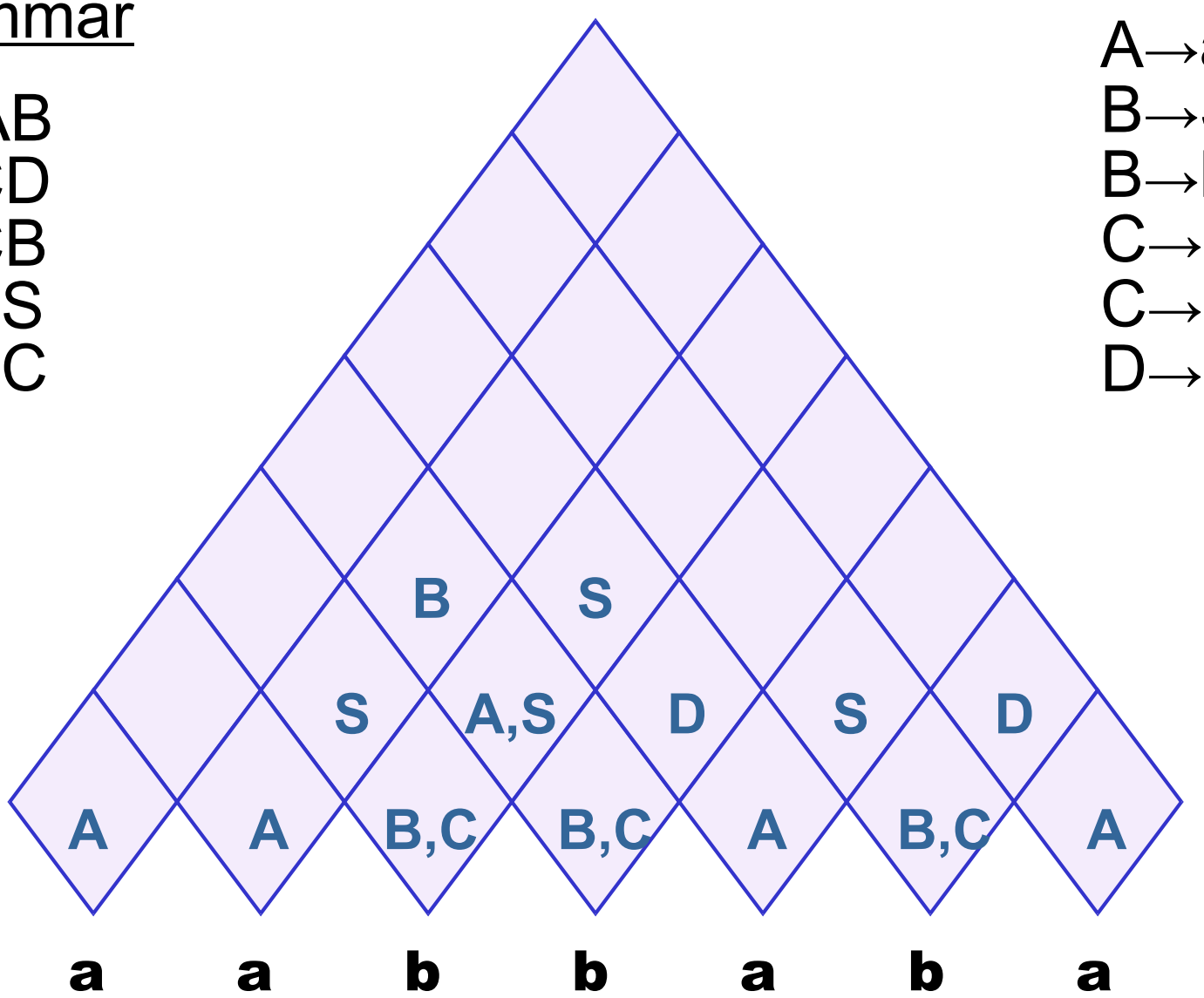
A→a
B→SC
B→b
C→DD
C→b
D→BA



Grammar

S→AB
S→CD
S→CB
S→SS
A→BC

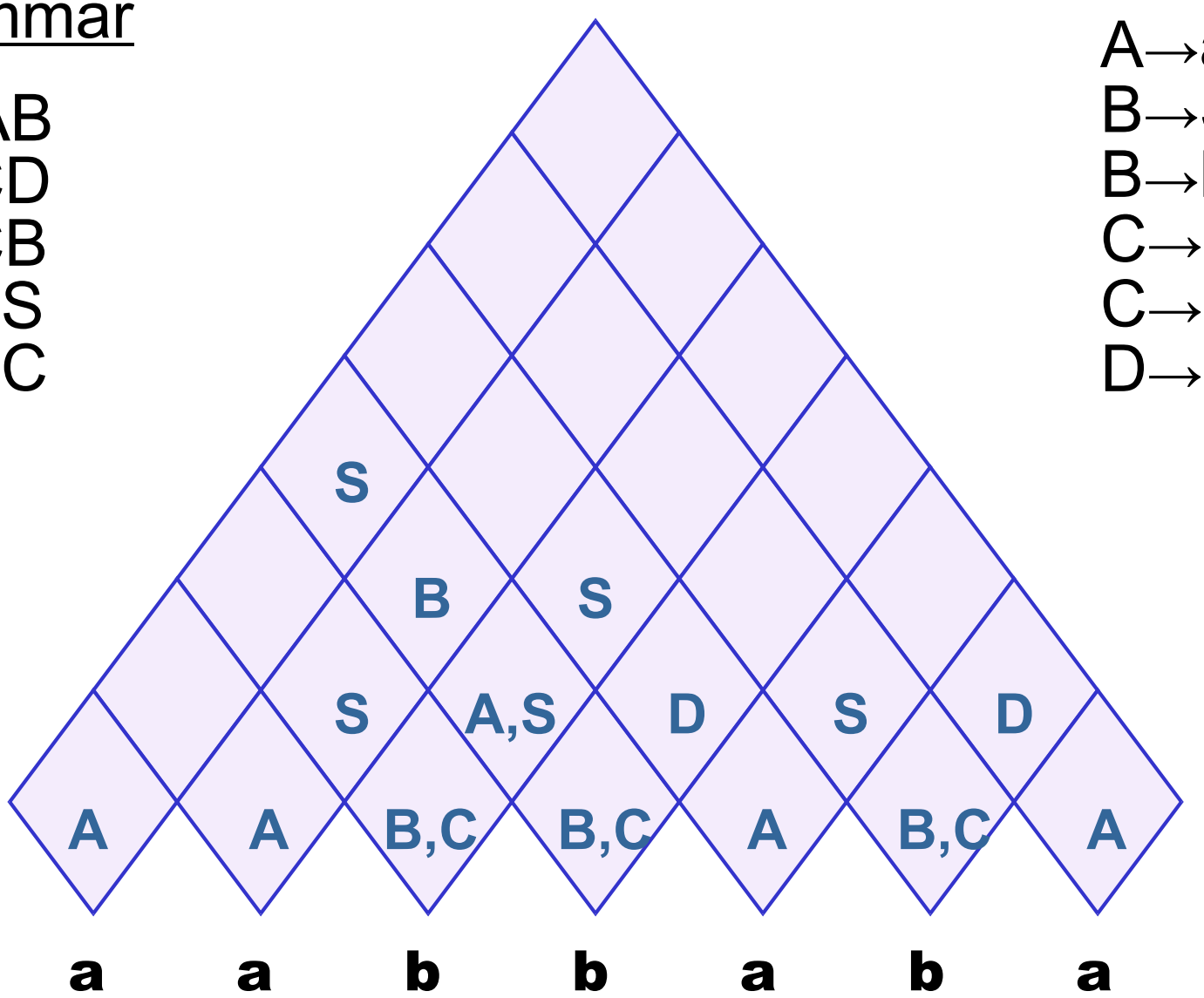
A→a
B→SC
B→b
C→DD
C→b
D→BA



Grammar

S→AB
S→CD
S→CB
S→SS
A→BC

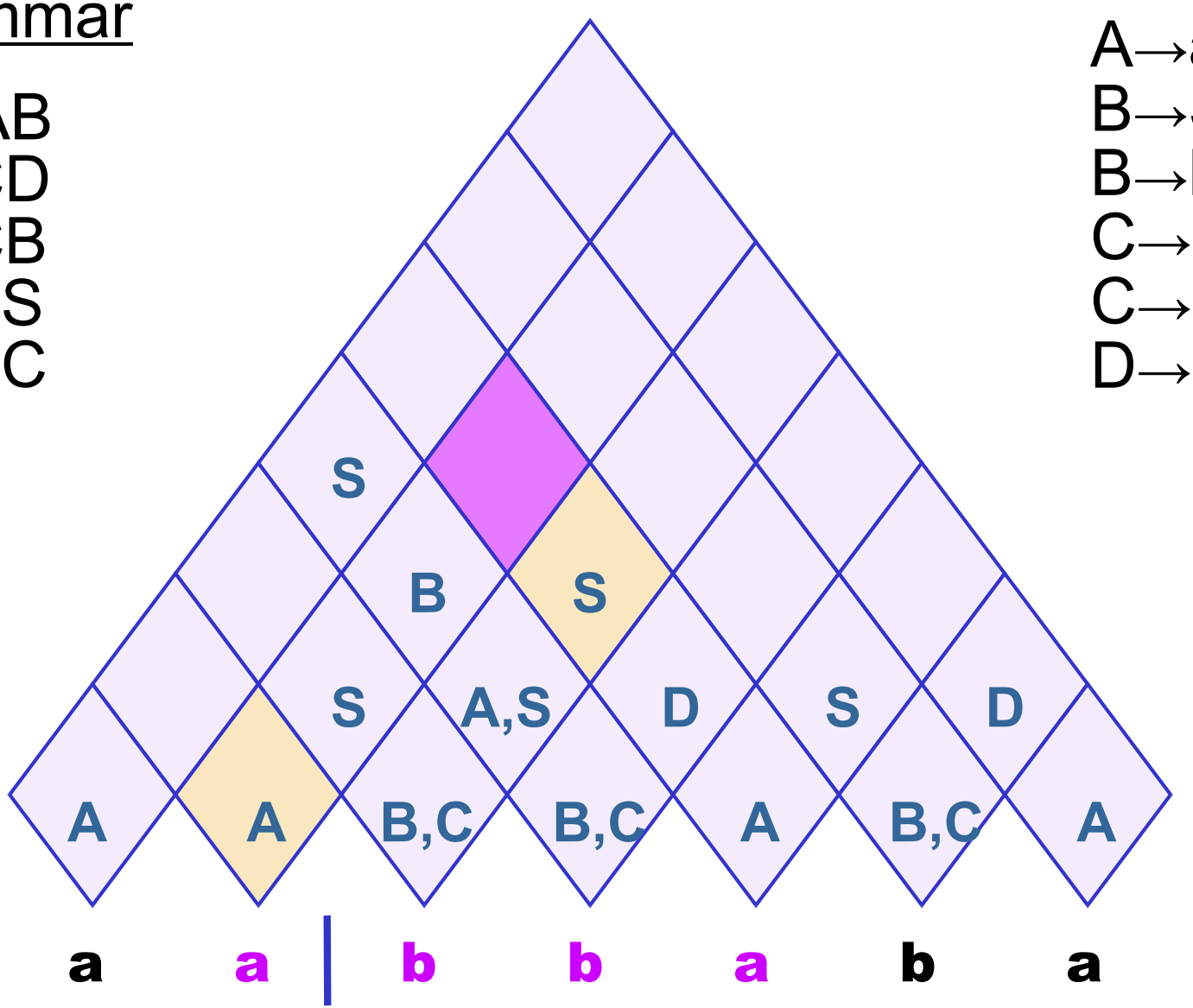
A→a
B→SC
B→b
C→DD
C→b
D→BA



Grammar

S→AB
S→CD
S→CB
S→SS
A→BC

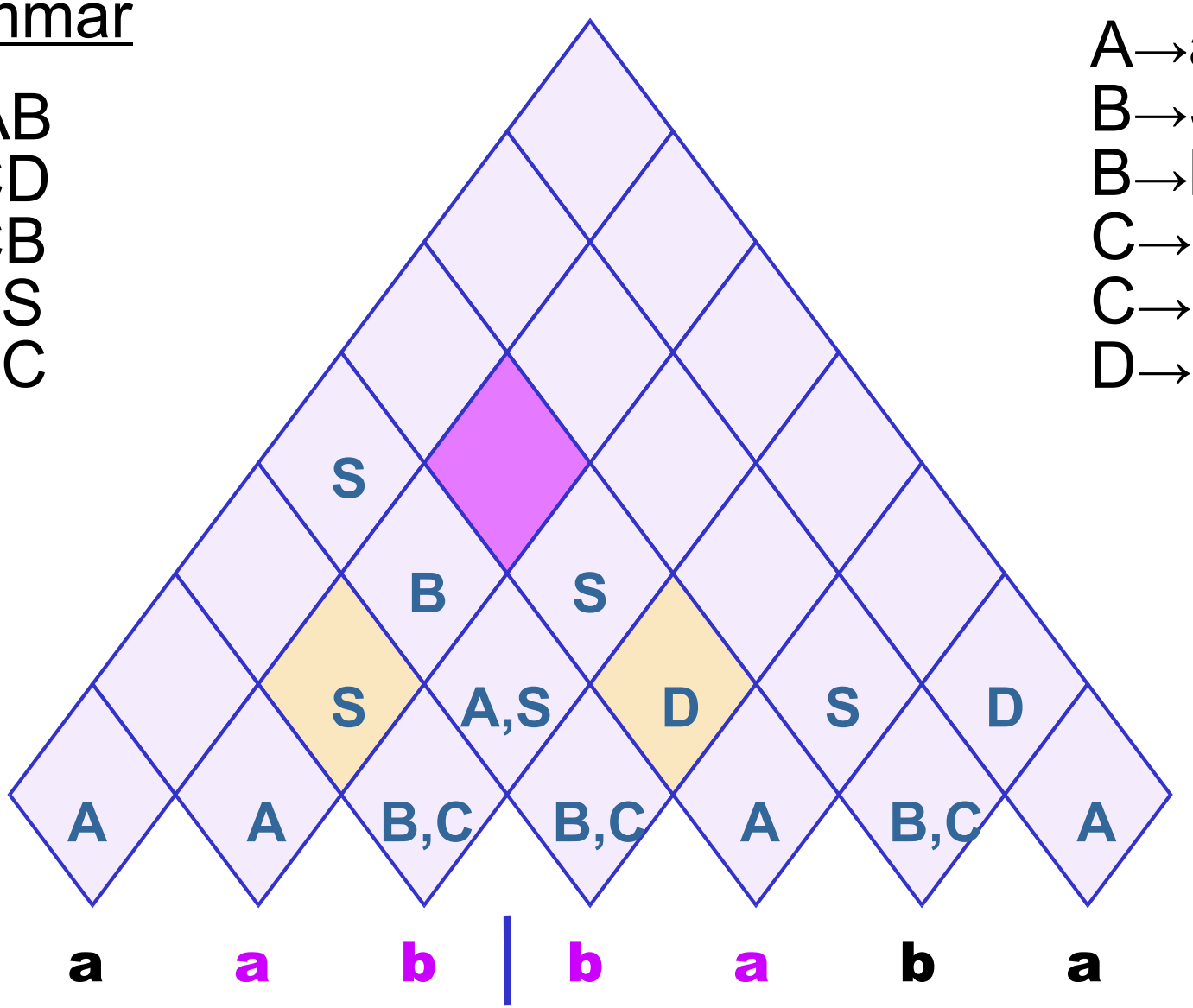
A→a
B→SC
B→b
C→DD
C→b
D→BA



Grammar

S→AB
S→CD
S→CB
S→SS
A→BC

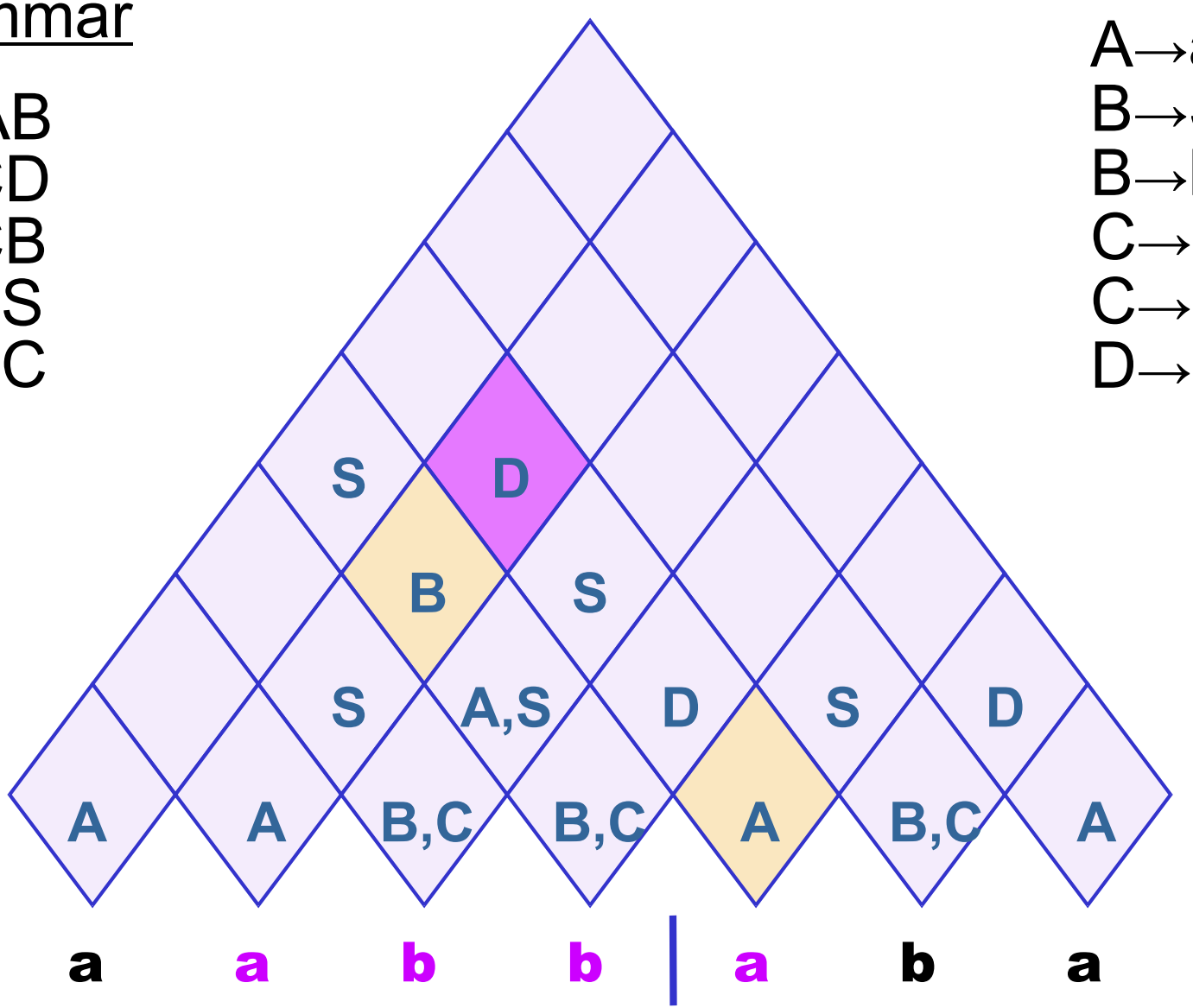
A→a
B→SC
B→b
C→DD
C→b
D→BA



Grammar

S→AB
S→CD
S→CB
S→SS
A→BC

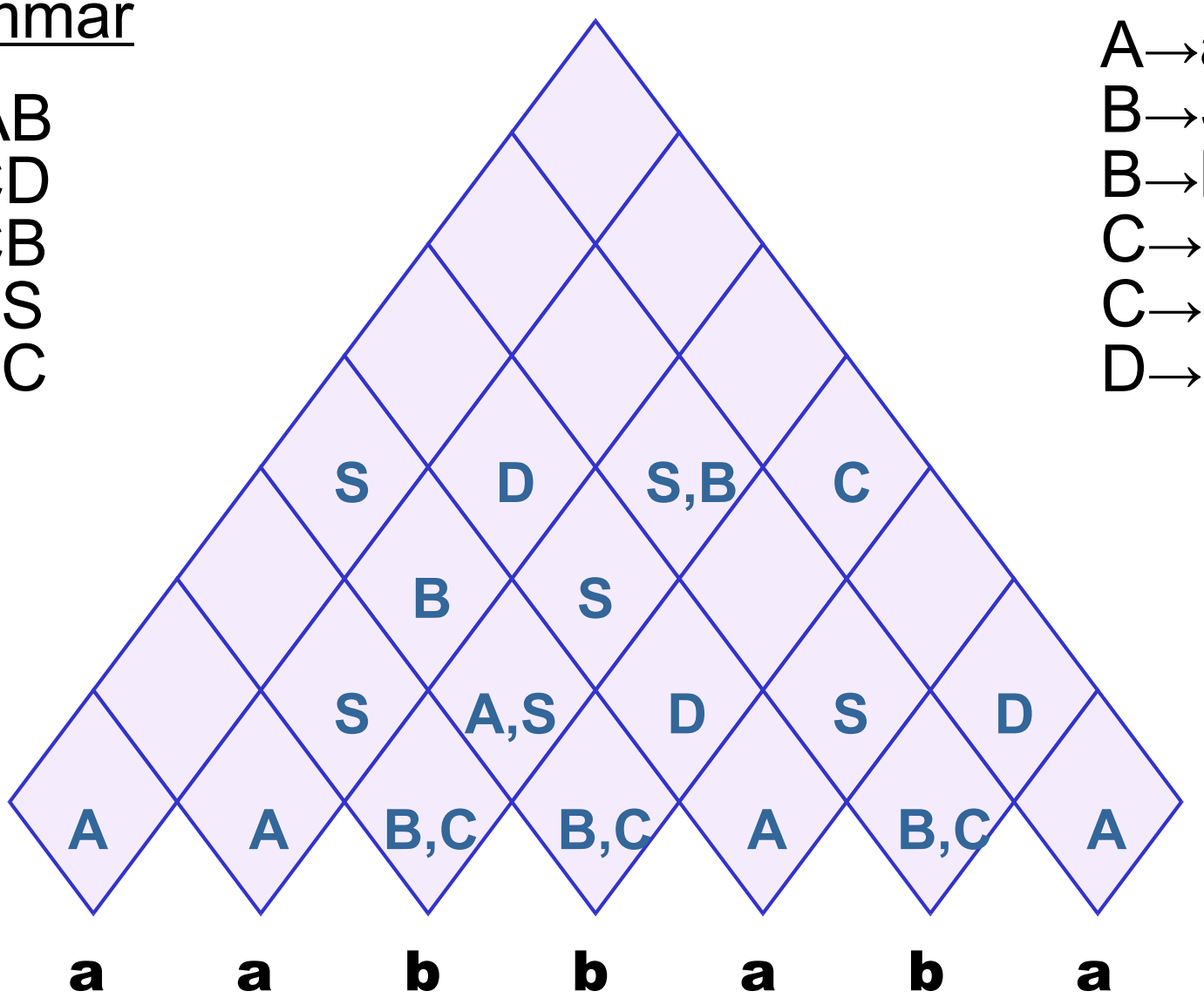
A→a
B→SC
B→b
C→DD
C→b
D→BA



Grammar

S→AB
S→CD
S→CB
S→SS
A→BC

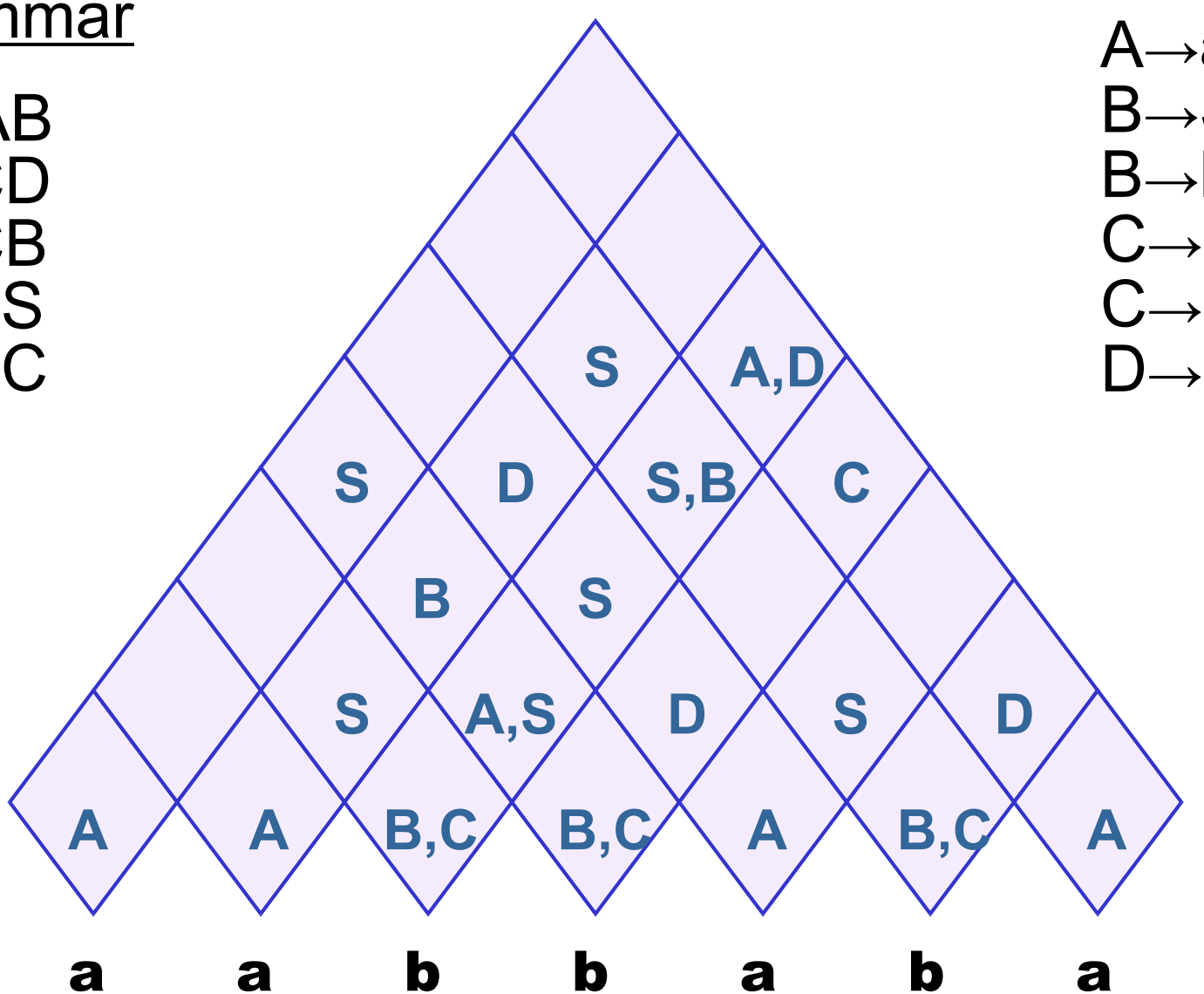
A→a
B→SC
B→b
C→DD
C→b
D→BA



Grammar

S→AB
S→CD
S→CB
S→SS
A→BC

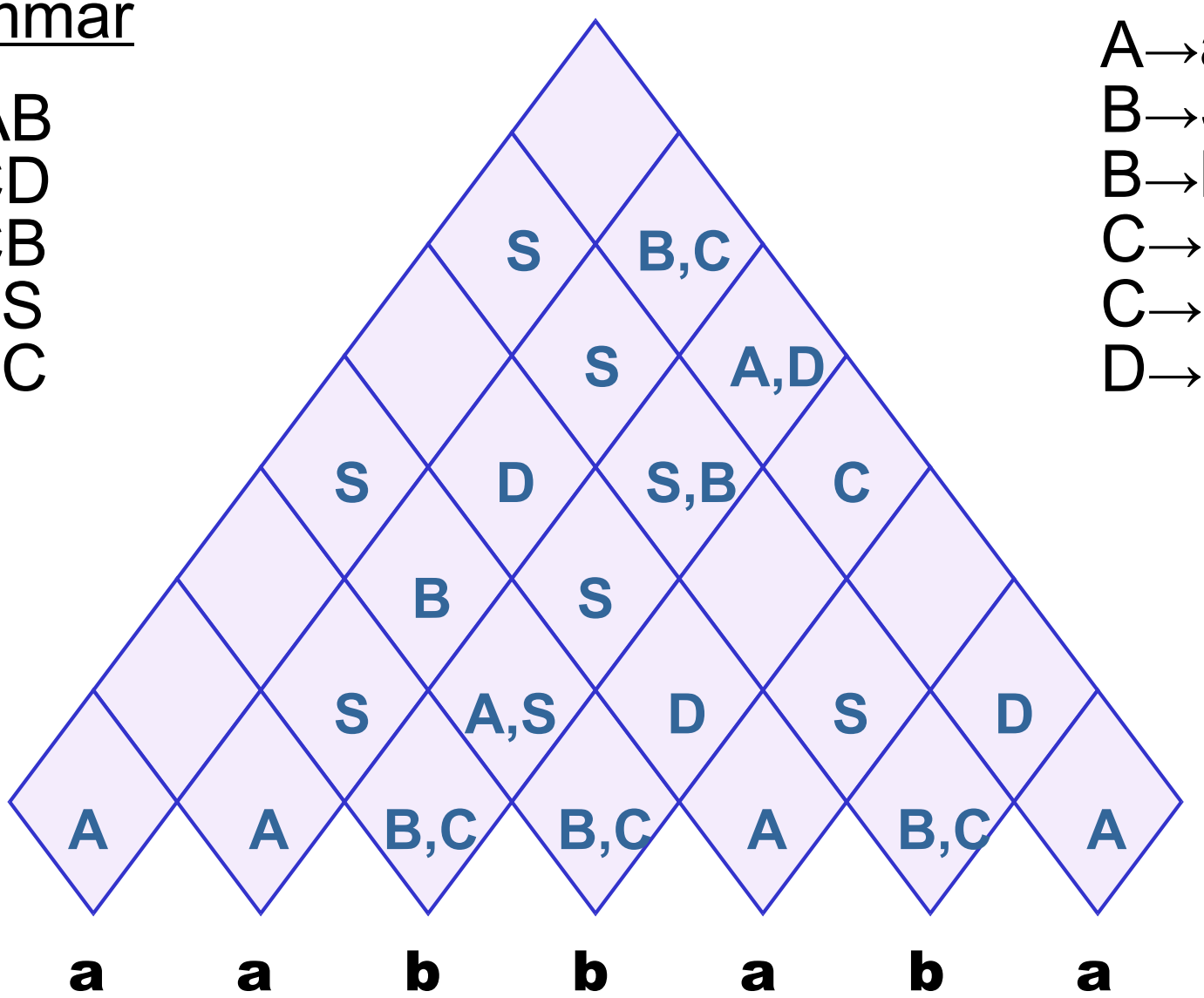
A→a
B→SC
B→b
C→DD
C→b
D→BA



Grammar

S→AB
S→CD
S→CB
S→SS
A→BC

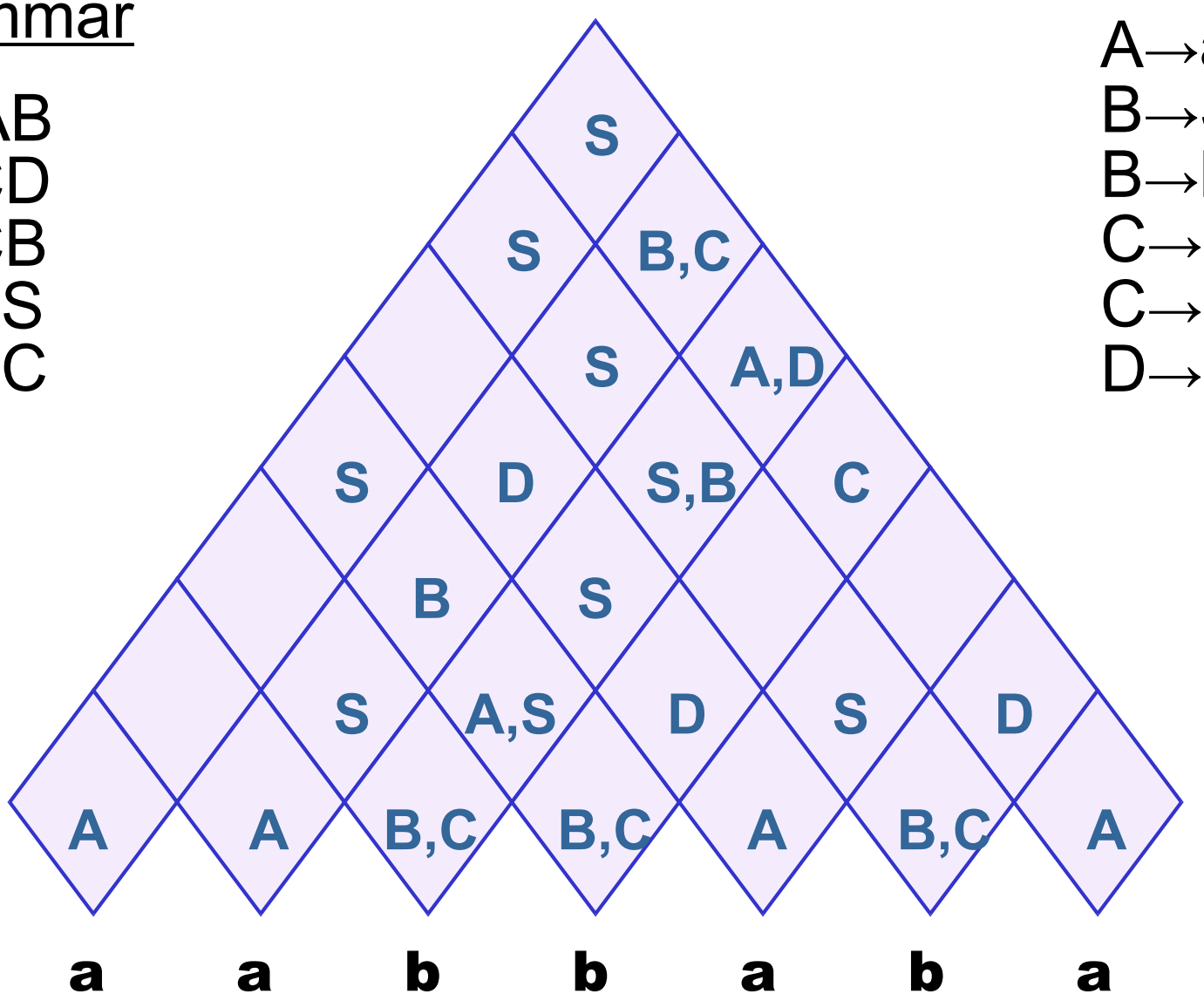
A→a
B→SC
B→b
C→DD
C→b
D→BA

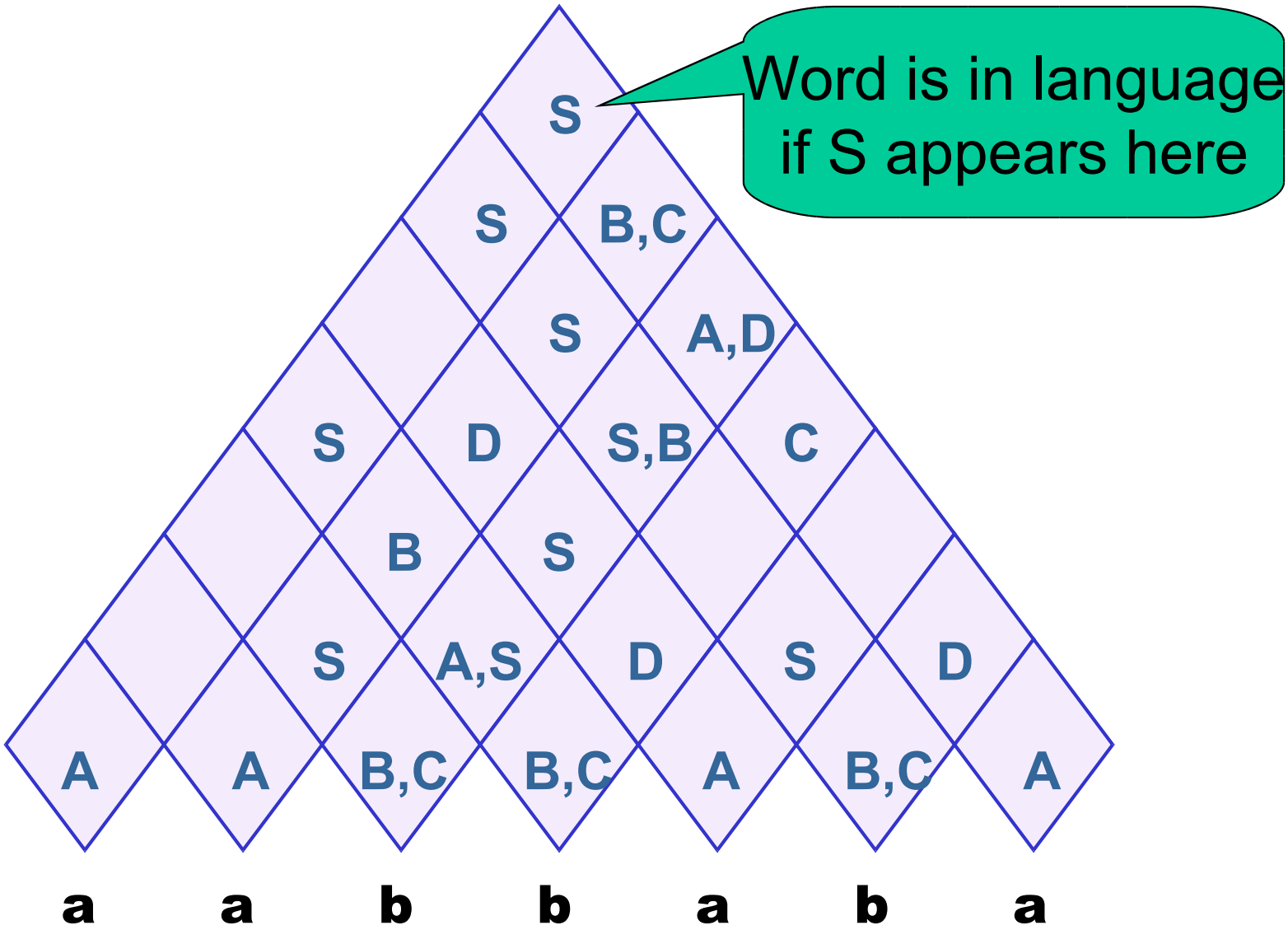


Grammar

S→AB
S→CD
S→CB
S→SS
A→BC

A→a
B→SC
B→b
C→DD
C→b
D→BA




Word is in language if S appears here
S
S B,C
S A,D
S D S,B C
B S
S A,S D S D
A A B,C B,C A B,C A
a a b b a b a

Revisit complexity

exercise

Parse “baaba” for this grammar

S -> AB | BC
A -> BA | a
B -> CC | b
C -> AB | a

(Hopcroft, Motwani, Ullman, p301)