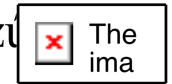
A CYK algoritmus (Cocke-Younger-Kasami)

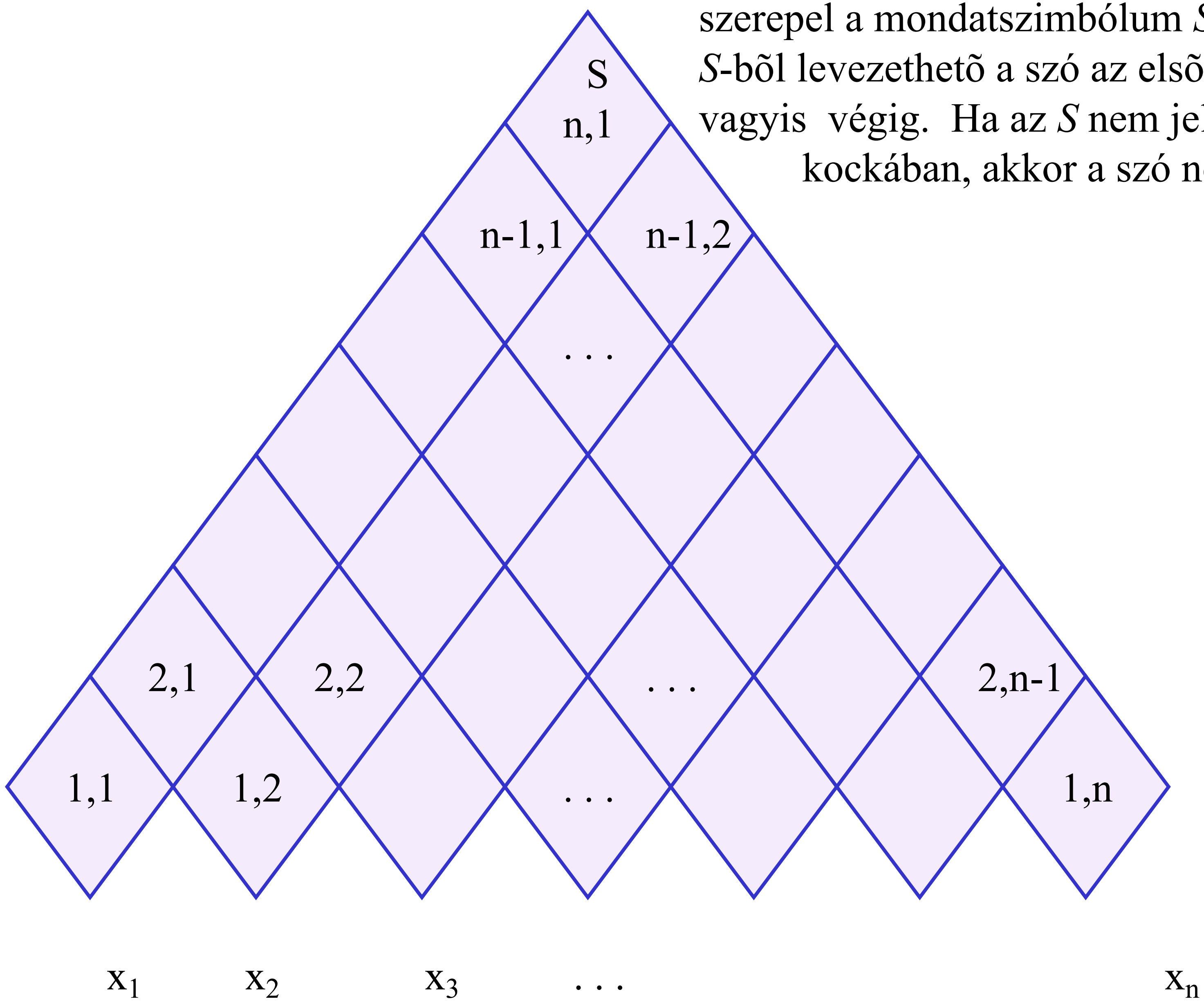
Az algoritmussal tetszőleges Chomsky féle normál alakban megadott nyelvtan és tetszőleges terminális sztring esetén eldönthető (polinomiális időben), hogy a sztring eleme-e a nyelvtan által generált nyelvnek.

Az algoritmus egy alulról felfele történõ elemzést valósít meg. Ahhoz, hogy működjön az kell, hogy a G=(V_N, V_T , S, H) nyelvtan Chomsky normál alakban (CNF) legyen, azaz a nyelvtanban csak $A \rightarrow BC$ ($A, B, C \in V_N$), illetve $A \rightarrow a$ ($A \in V_N$, $a \in V_T$) alakú szabályok vannak . Ha egy $n > 0$ hosszú 

($x_1, \ldots, x_n \in V_T$) szót szeretnénk elemezni, akkor egy n x n-es alsó háromszög mátrix alakú táblázatot fogunk kitölteni a következõ módon. A sorokat alulról felfele számozzuk, az oszlopokat balról jobbra. Az *i*. sor *j*. kockájába akkor kerül egy $A \in V_N$ nemterminálisa a nyelvtannak, ha az *A*-ból levezethetõ az elemzendõ input szó azon darabkája, ami a *j*. betûnél kezdõdik és *i* hosszan tart, vagyis levezethetõ az 

szszó.

Ha kitöltjük a táblázatot és a legfelsõ mezõben szerepel a mondatszimbólum *S* az azt jelenti, hogy *S*-bõl levezethetõ a szó az elsõ betûtõl az *n*.-ig, vagyis végig. Ha az *S* nem jelenik meg a legfelsõ kockában, akkor a szó nem eleme a nyelvnek.



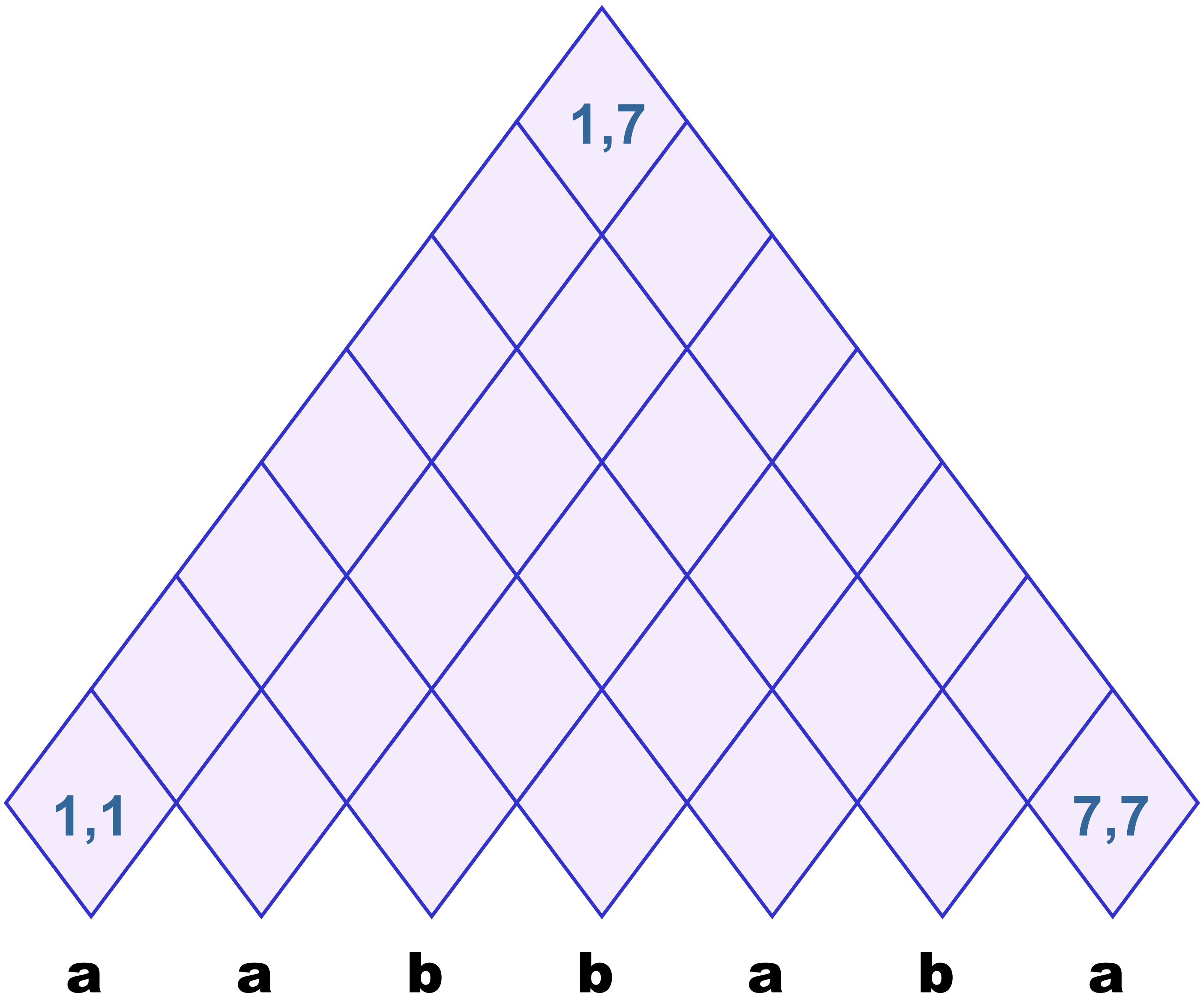
A táblázat kitöltése:

Az elsõ sor egyértelmû: azok a nemterminálisok kerülnek a *k*. mezõbe, akik egy lépésben a *k*. terminálist generálják egy alakú szabállyal.

Késõbbi sorok: egy *A* akkor lesz az *i*. sor *j*. oszlopában, ha belõle levezethetõ az szó. Mivel csak alakú szabályok vannak ezért ez csak úgy lehet, hogy a *B* megcsinálja -t (az elejét, valameddig), a *C* pedig -et (a maradékot). De ezt már le lehet ellenõrizni, mert ezek az információk a táblázat már kitöltött részében benne vannak. Tehát egy *A*-t akkor írunk be az *i*. sor *j*. kockájába, ha van olyan szabály, hogy *B* benne van a *j*. oszlop *k*. sorában valami *k*-ra, a *C* meg benne van a $\boxed{k+j}$. oszlop *i*-*k*. sorában.

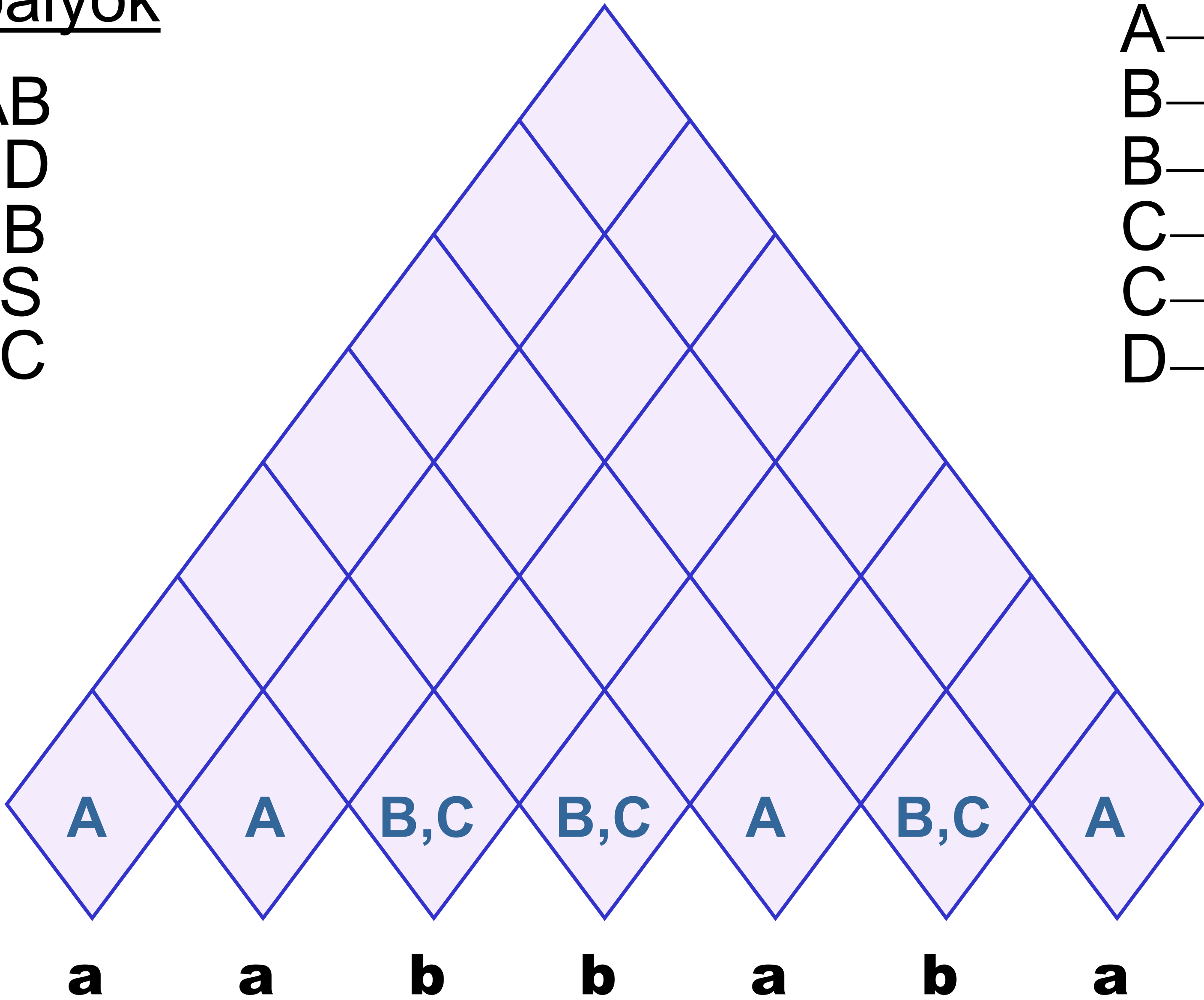
Ha nem csak arra vagyunk kíváncsiak, hogy generálni lehet-e a szót, hanem arra is, hogy hogyan, akkor nem csak a megfelelõ nemterminálist írjuk be a táblázatba, hanem ellátjuk két indexszel is: az elsõ mutatja, hogy milyen felbontásban generálja a *BC* sorozat a szórészletet (azaz, hogy a *B* hány darab betû generál, ez a fenti jelölésekkel a *k*), a második meg annak a szabálynak a száma, amit használunk (vagyis az szabály sorszáma, a szabályokat még az elején megszámoztuk, hogy lehessen rájuk hivatkozni). Az elsõ index tulajdonképpen azt mutatja, hogy az így beírt *A* oszlopában hányadik sorban kell keresnünk a *B*-t, a szabály száma meg azt mutatja, hogy mit is kell keresnünk. Így a levezetési fa felépíthetõ. Ha ezen visszakeresés során elágazást tapasztalunk (azaz van olyan kocka, ahol két ugyanolyan, de más indexû nemterminális áll), akkor a szó nem egyértelmûen áll elõ. Ekkor a visszakeresõs eljárás mindkét levezetési fát megadja.

Példa a C-Y-K-algoritmus alkalmazására


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

Szabályok

$S \to AB$
$S \to CD$
$S \to CB$
$S \to SS$
$A \to BC$

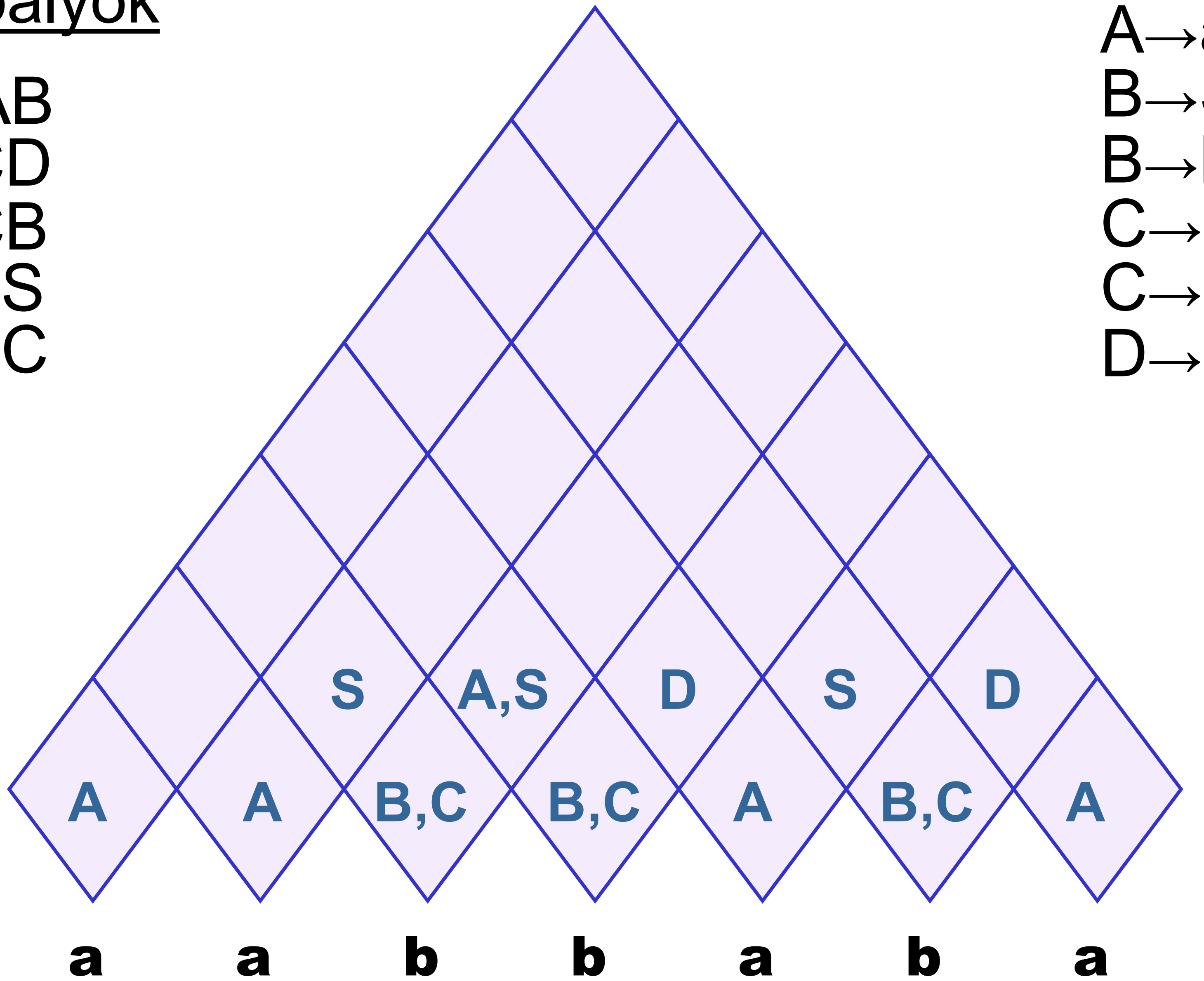


$A \to a$
$B \to SC$
$B \to b$
$C \to DD$
$C \to b$
$D \to BA$

Szabályok

$S \to AB$
$S \to CD$
$S \to CB$
$S \to SS$
$A \to BC$

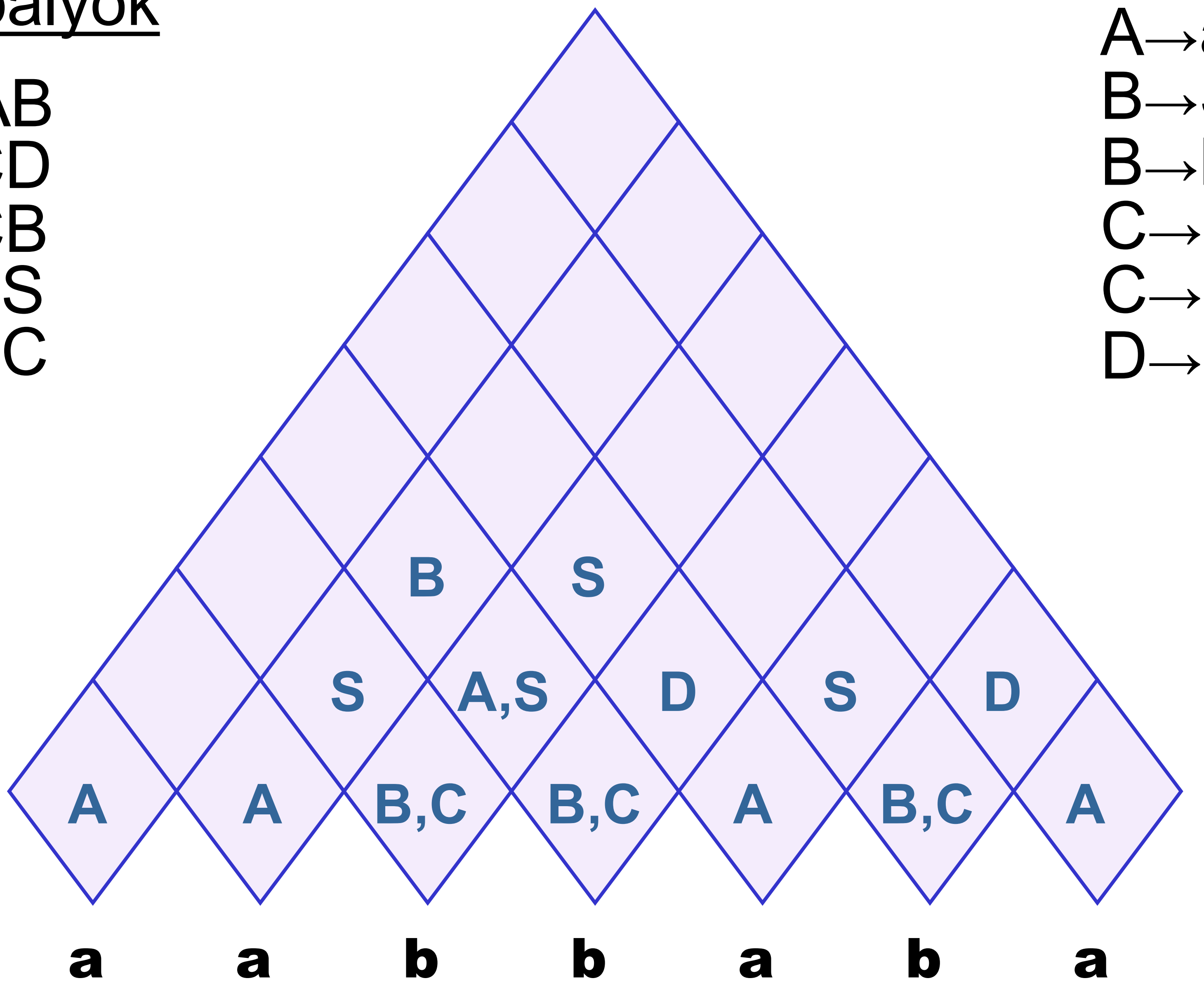
$A \to a$
$B \to SC$
$B \to b$
$C \to DD$
$C \to b$
$D \to BA$



Szabályok

S→AB
S→CD
S→CB
S→SS
A→BC

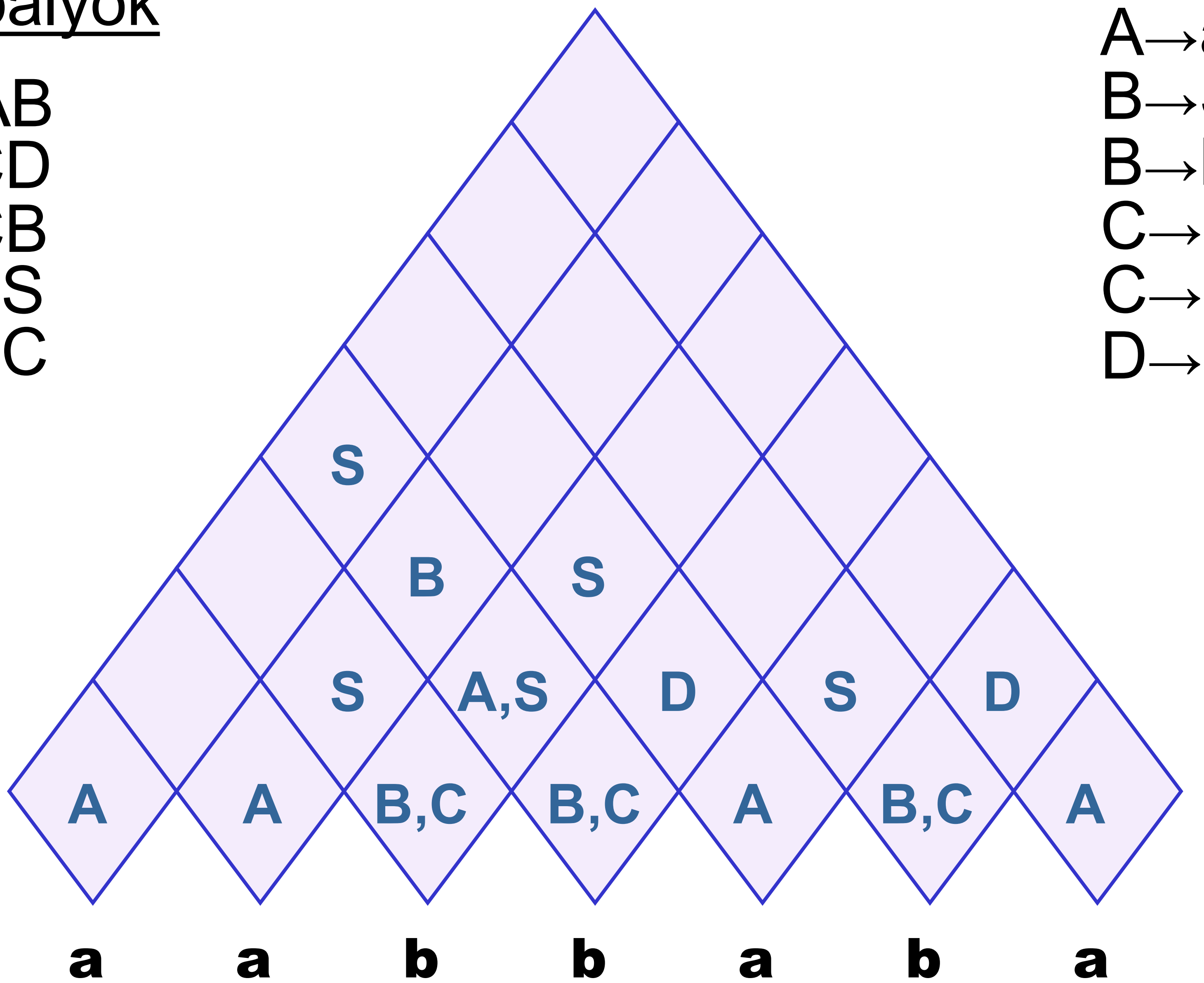
A→a
B→SC
B→b
C→DD
C→b
D→BA



Szabályok

S→AB
S→CD
S→CB
S→SS
A→BC

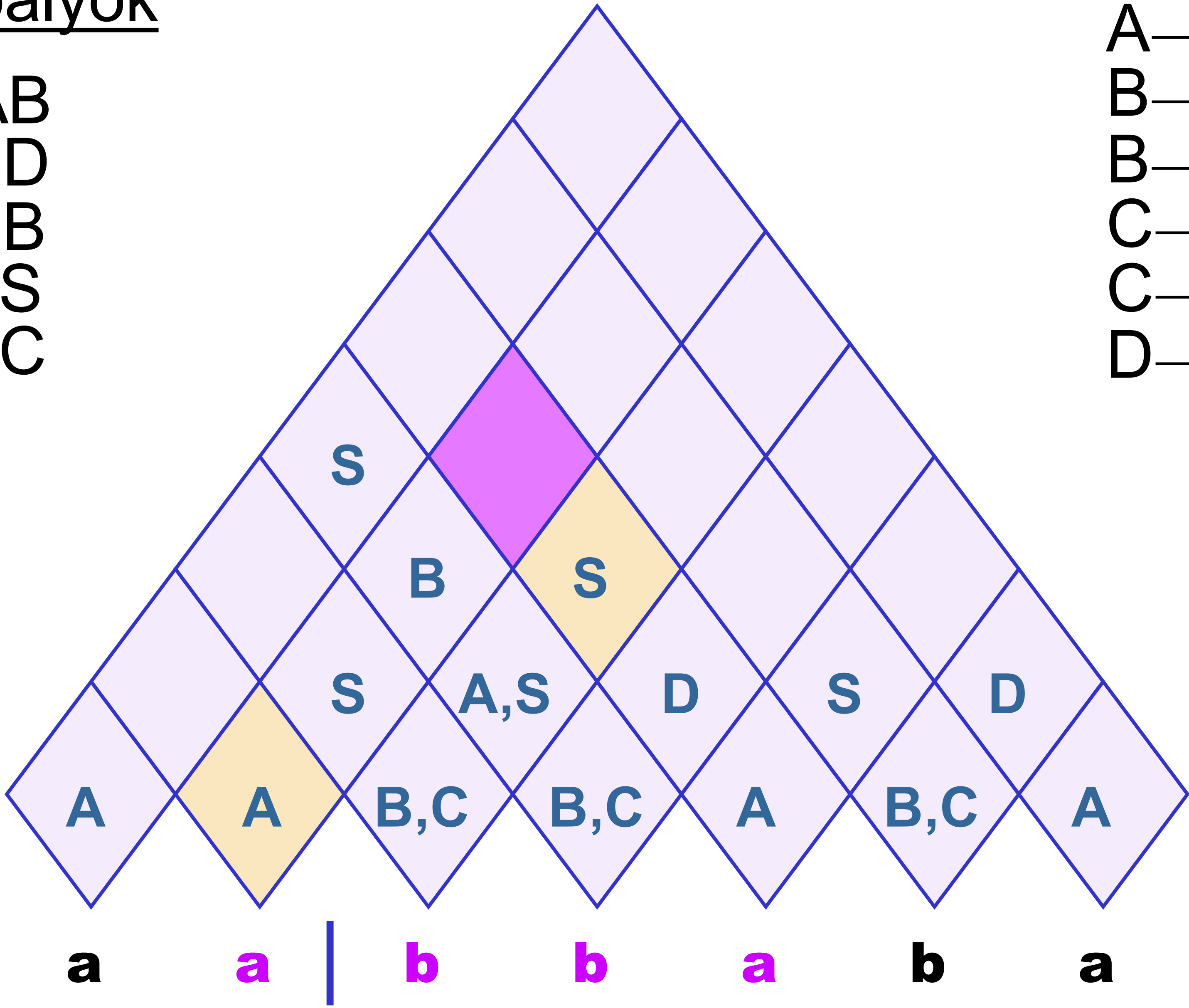
A→a
B→SC
B→b
C→DD
C→b
D→BA



<u>Szabályok</u>

S→AB
S→CD
S→CB
S→SS
A→BC

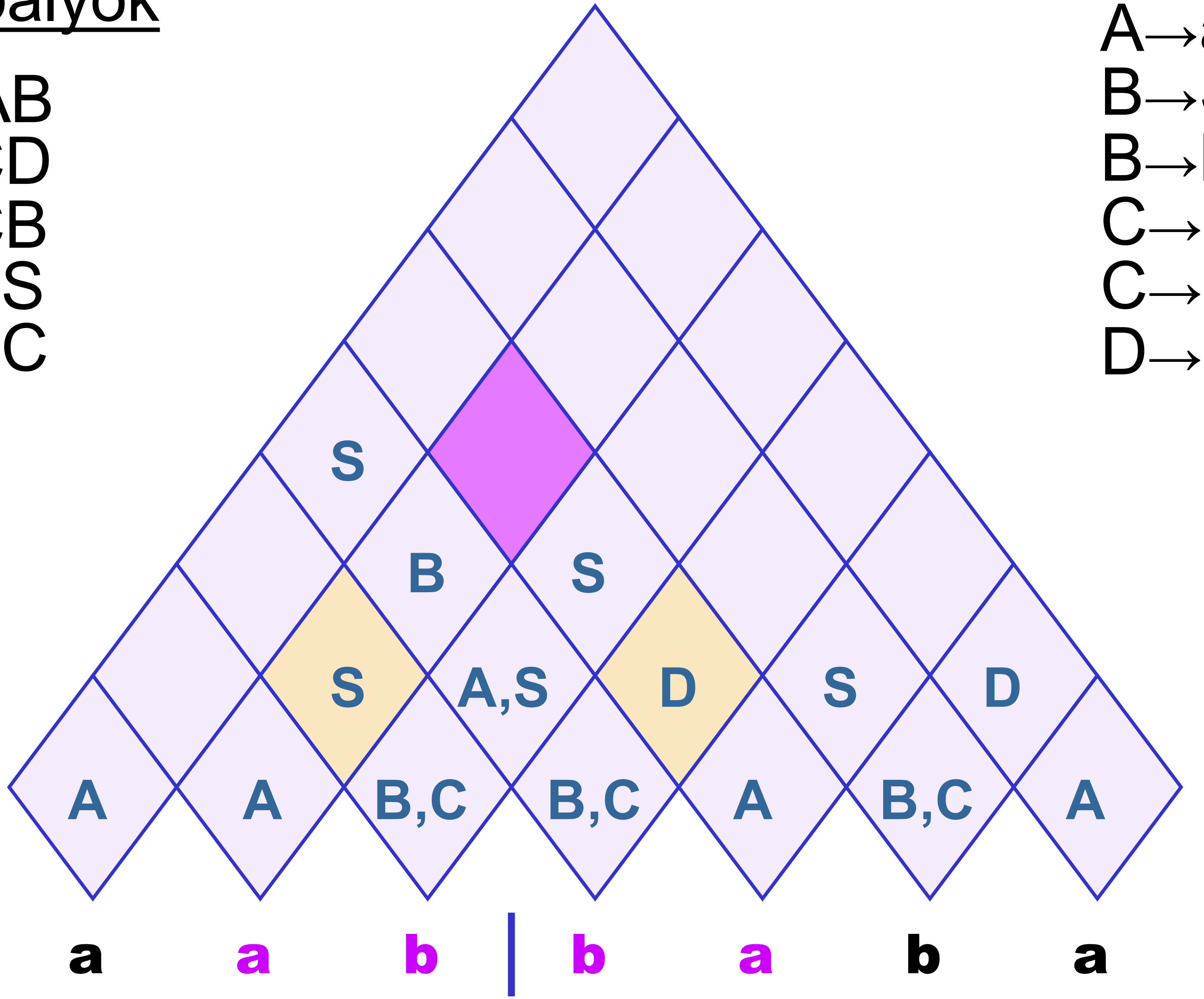
A→a
B→SC
B→b
C→DD
C→b
D→BA



Szabályok

S→AB
S→CD
S→CB
S→SS
A→BC

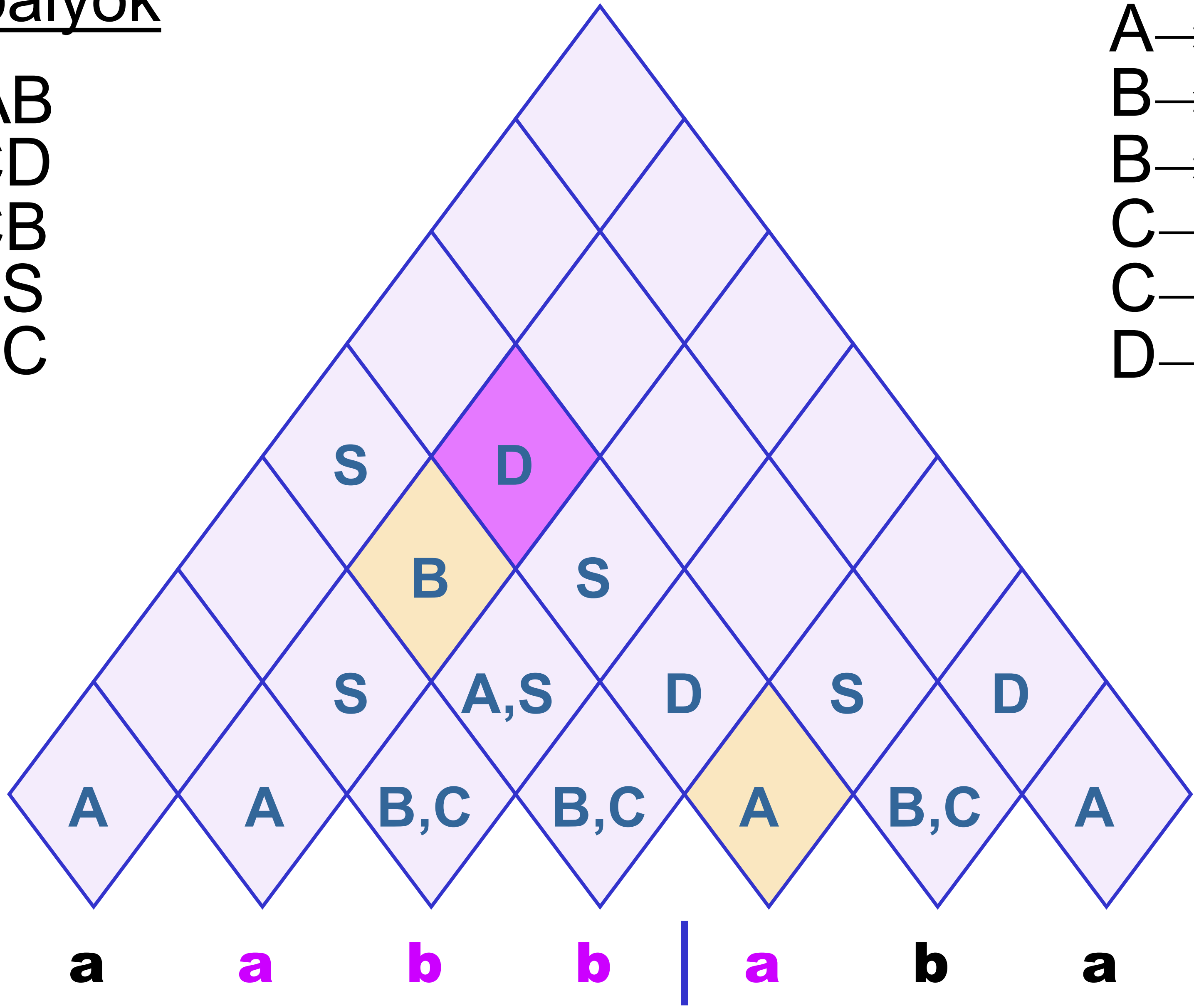
A→a
B→SC
B→b
C→DD
C→b
D→BA



Szabályok

$S \rightarrow AB$
$S \rightarrow CD$
$S \rightarrow CB$
$S \rightarrow SS$
$A \rightarrow BC$

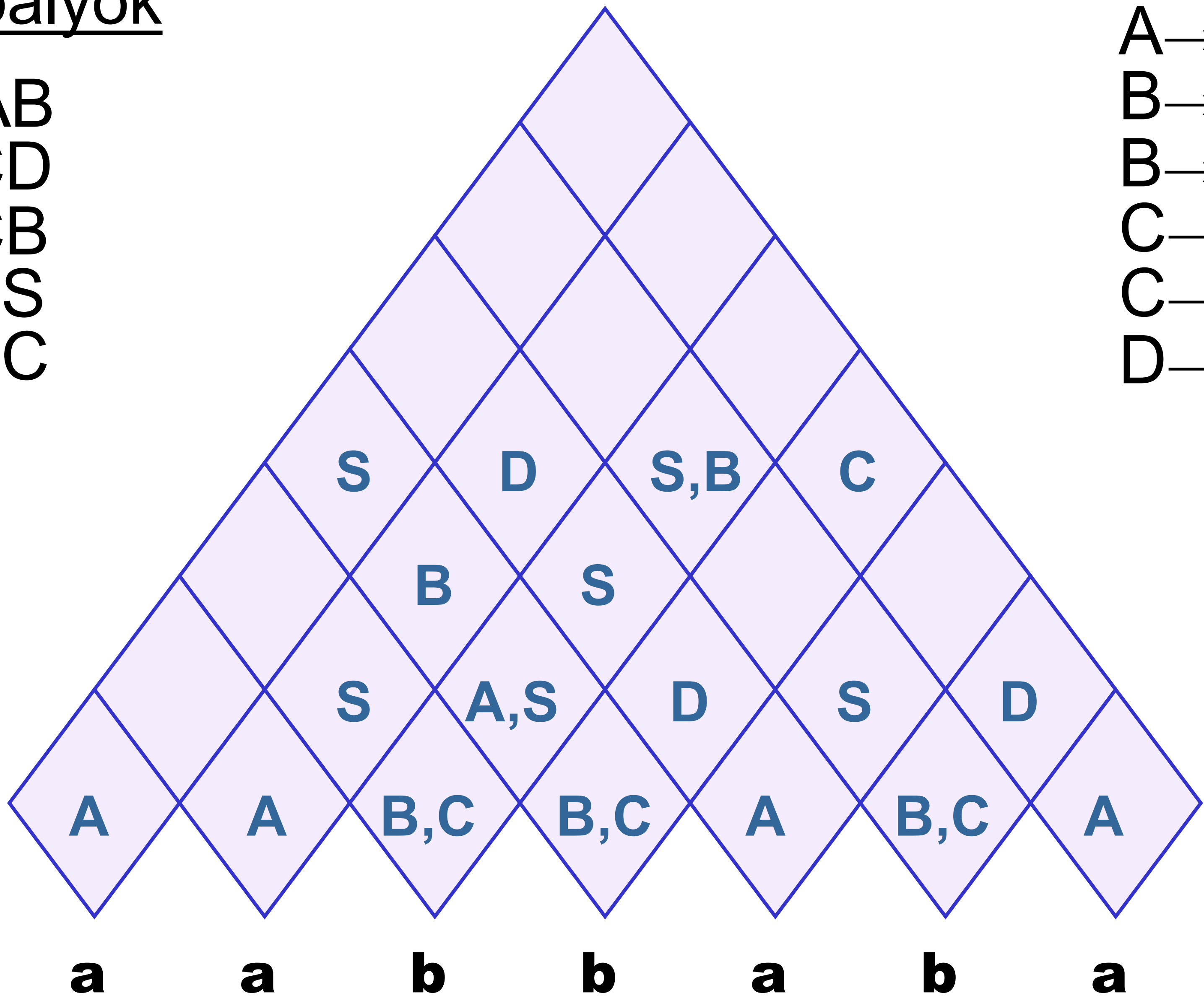
$A \rightarrow a$
$B \rightarrow SC$
$B \rightarrow b$
$C \rightarrow DD$
$C \rightarrow b$
$D \rightarrow BA$



Szabályok

$S \to AB$
$S \to CD$
$S \to CB$
$S \to SS$
$A \to BC$

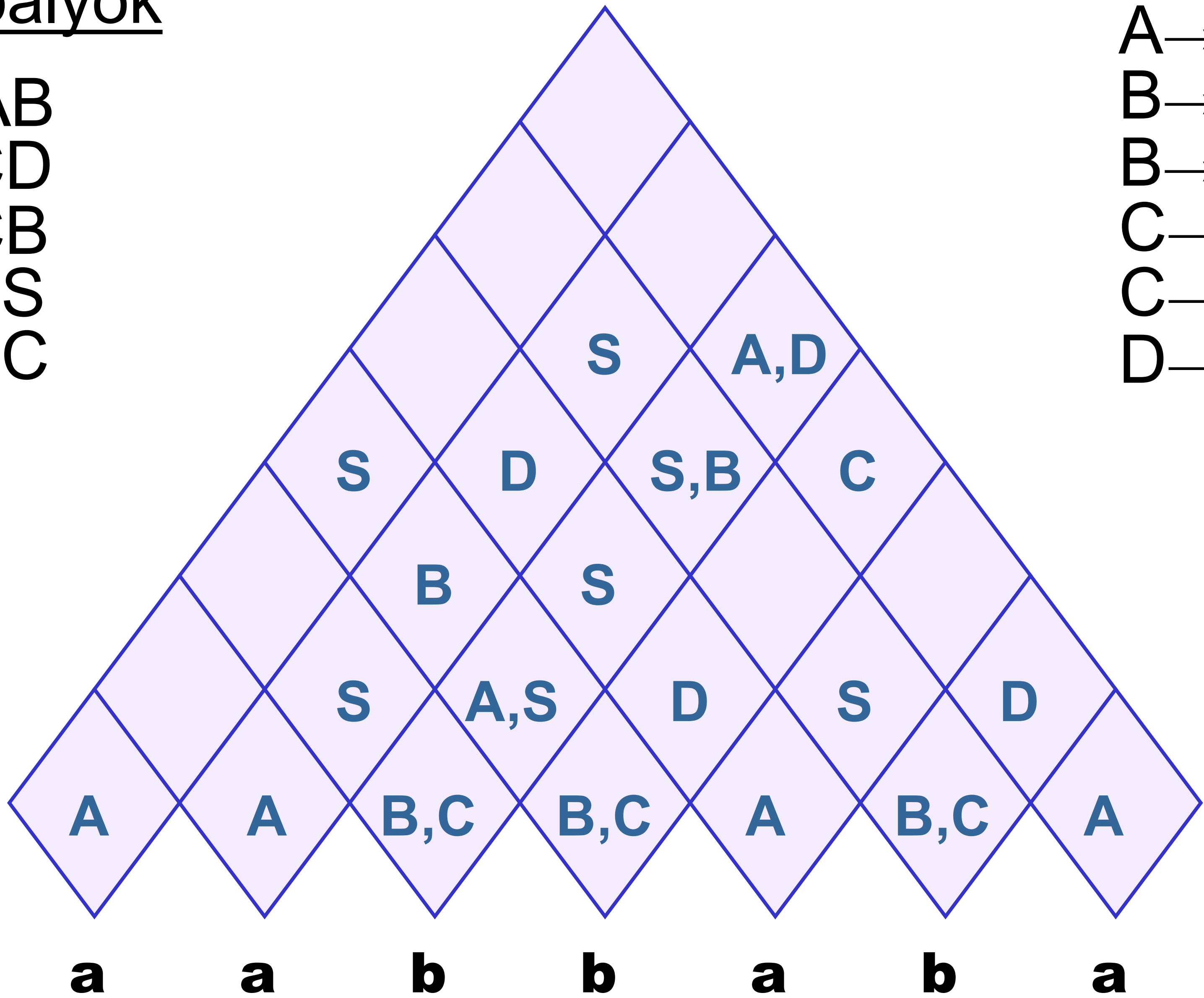
$A \to a$
$B \to SC$
$B \to b$
$C \to DD$
$C \to b$
$D \to BA$



Szabályok

$S \to AB$
$S \to CD$
$S \to CB$
$S \to SS$
$A \to BC$

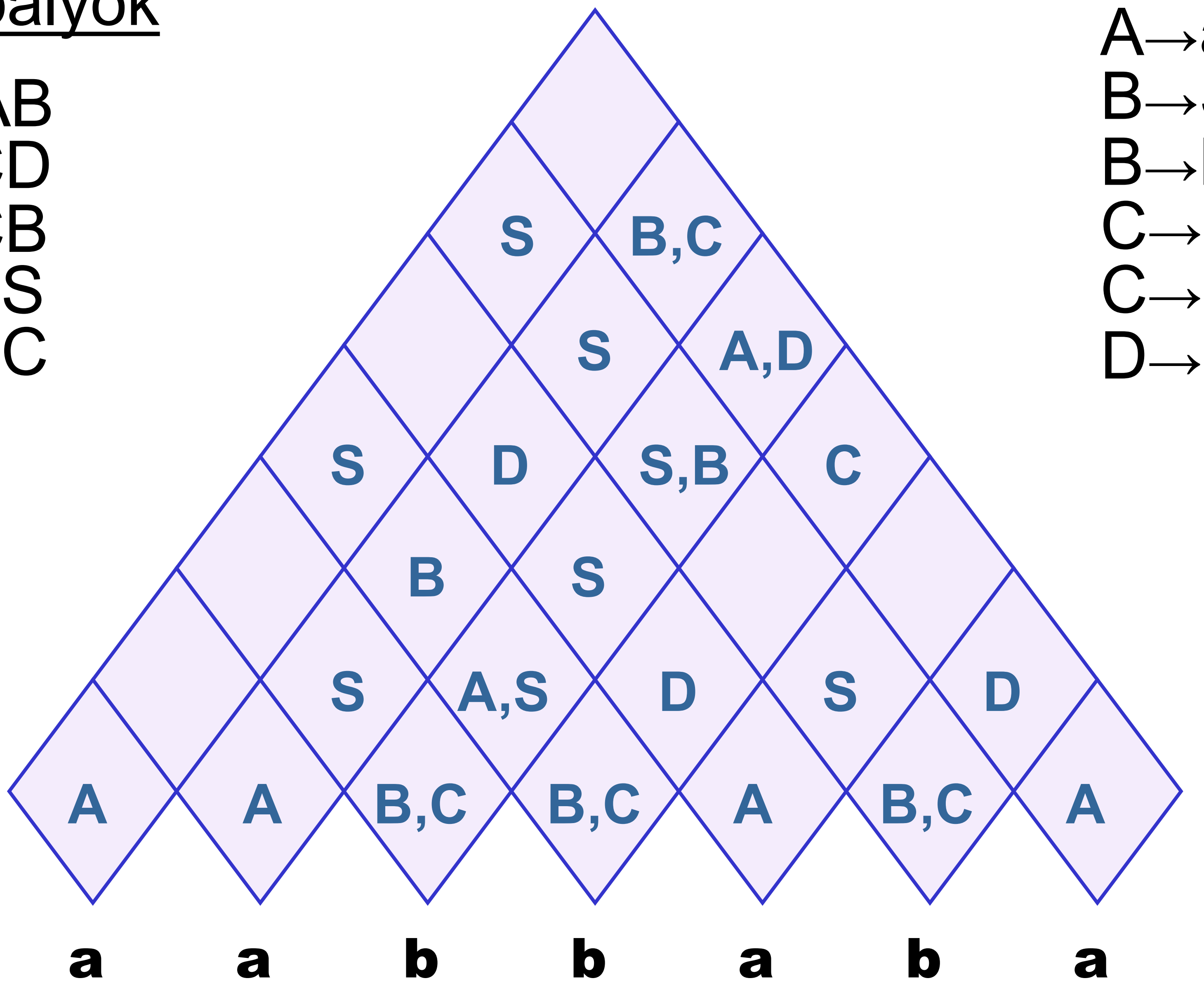
$A \to a$
$B \to SC$
$B \to b$
$C \to DD$
$C \to b$
$D \to BA$



Szabályok

$S \rightarrow AB$
$S \rightarrow CD$
$S \rightarrow CB$
$S \rightarrow SS$
$A \rightarrow BC$

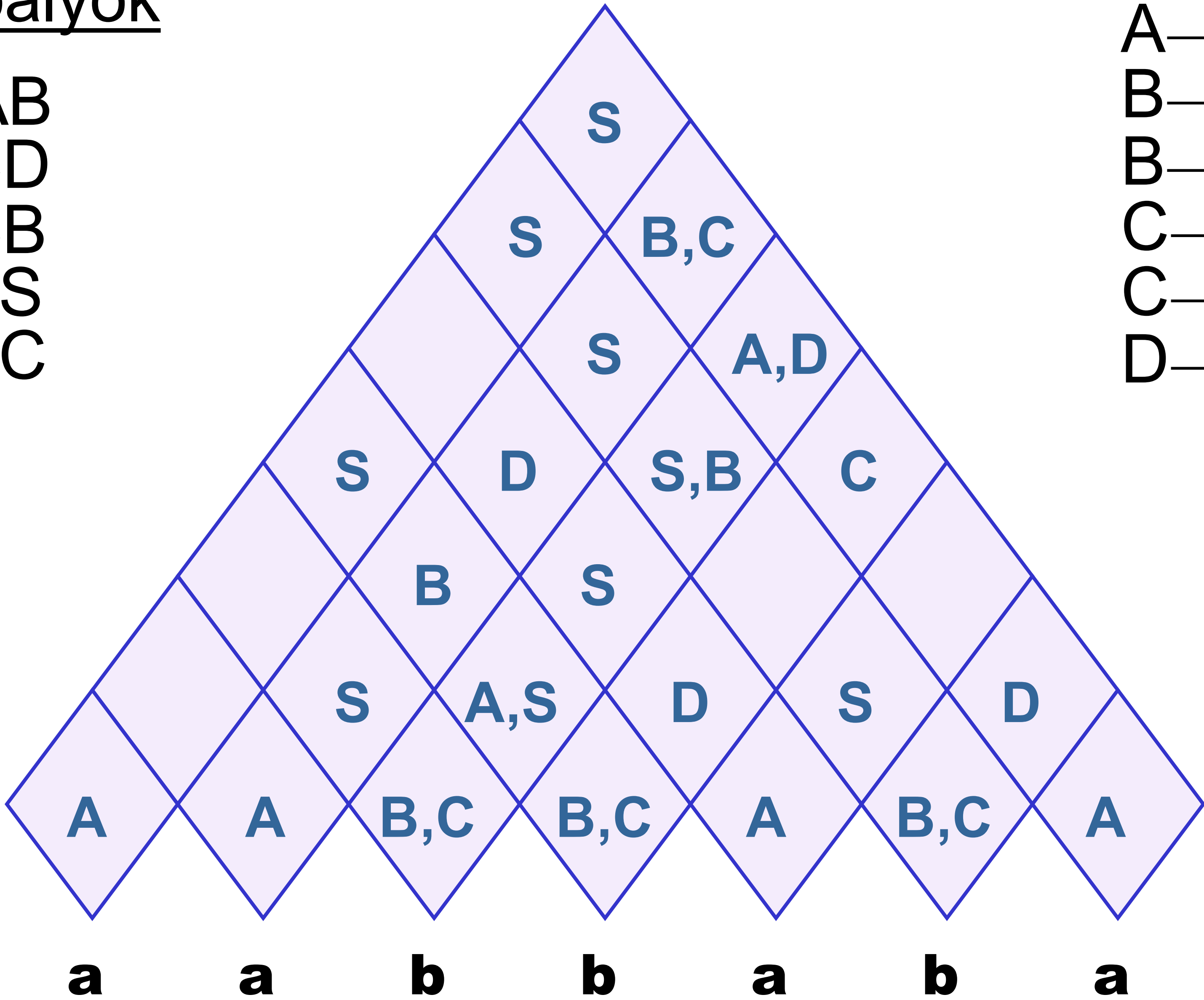
$A \rightarrow a$
$B \rightarrow SC$
$B \rightarrow b$
$C \rightarrow DD$
$C \rightarrow b$
$D \rightarrow BA$

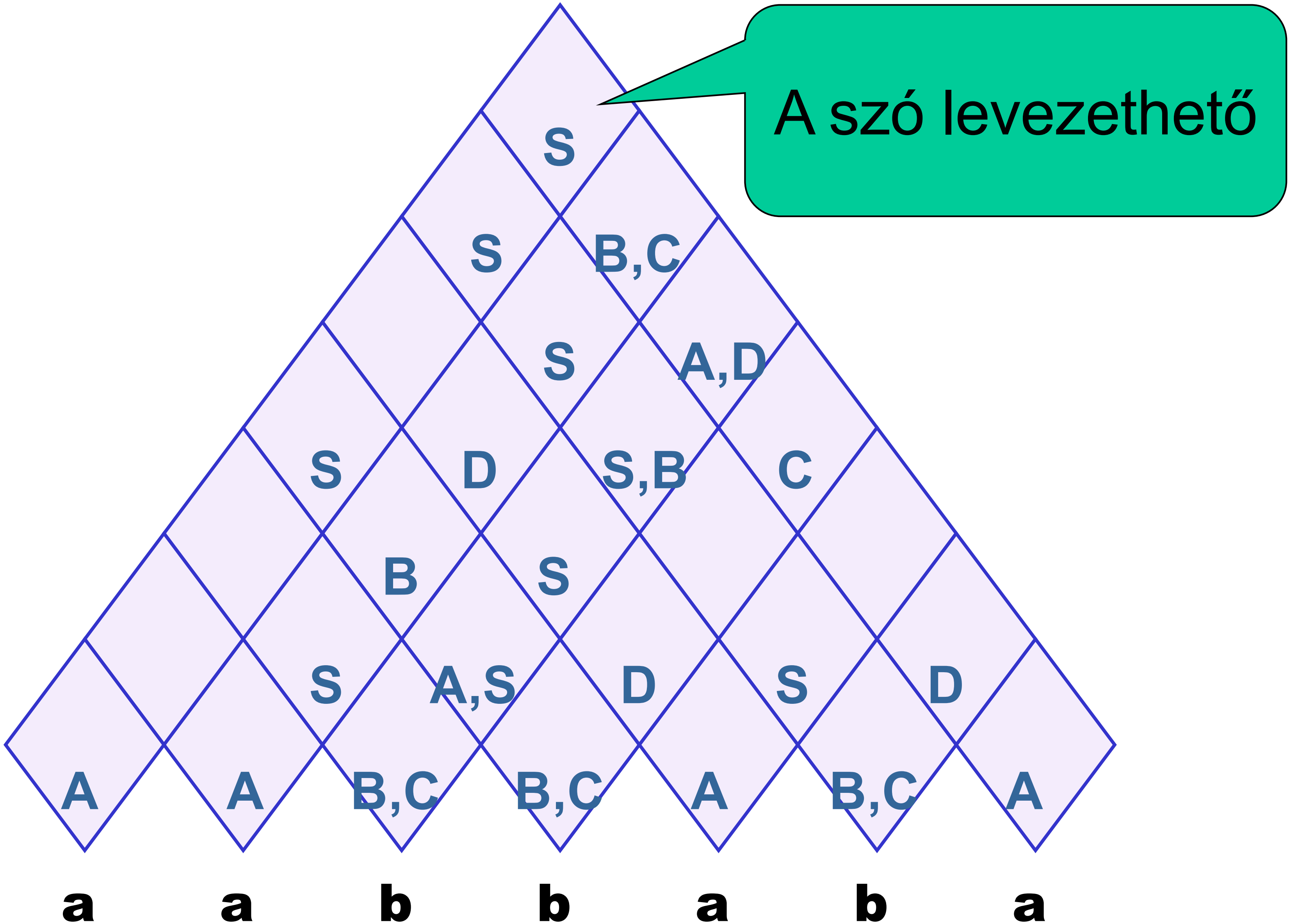


Szabályok

$S \to AB$
$S \to CD$
$S \to CB$
$S \to SS$
$A \to BC$

$A \to a$
$B \to SC$
$B \to b$
$C \to DD$
$C \to b$
$D \to BA$




A szó levezethető
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

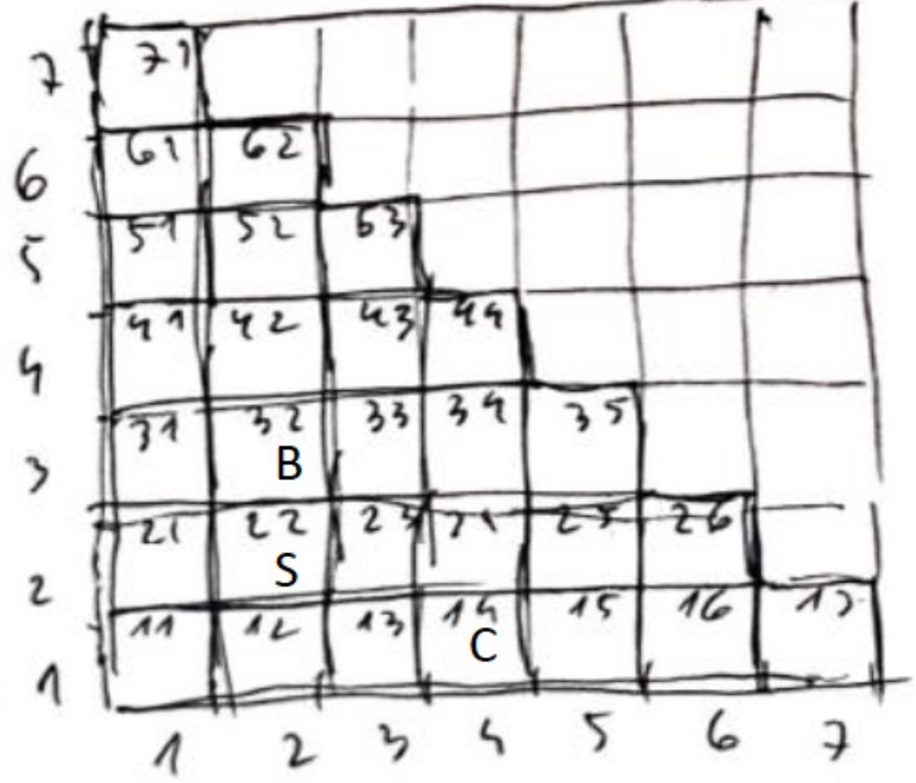
$\boxed{48}$ $\boxed{49}$ reciti számolásával,

$X \in i, j, \iff$ • $X \to YZ$

• $Y \in k, j$

$Z \in i-k, \boxed{k+j}$

valamely k-ra



$B \in 3,2$
• $B \to SC$
• $S \in 2,2$
$C \in 1,4$
$k=2$

bemenet: $a_1 a_2 .. a_n$, szabályok száma: t
G grammatika

```
for i = 1 .. n
    for j = 1 .. t
        ha a j-edik szabály jobb oldala a_i
           ~~baloldala~~ x
        akkor
            táblázat[1,i] =
                táblázat[1,i] ∪ a j-edik szabály bal oldala

for i = 2 ..
```

$X \in i,j \iff$ • $X \to YZ$
• $Y \in k,j$
$Z \in i-k, \boxed{k+j}$
valamilyen k-ra

for $i = 2..n$
  for $j = 1..(n-i+1)$
    for $k = 1..(i-1)$
      for $l = 1..t$
        ha az l.-edik szabály jobboldalának első betűje $\in$ táblázat[k,j]
        és
        az l-edik szabály jobboldalának második betűje $\in$ táblázat[$i-k$, $\boxed{k+j}$]
        akkor
        táblázat[i,j] = táblázat[i,j] $\cup$ az l-edik szabály bal oldala

ha $S \in$ táblázat[$n,1$]
  akkor $a_1 .. a_n \in L(G)$
  különben $a_1 .. a_n \notin L(G)$