T4P1E3b

Dado el lenguaje L = { $a^n b^m$ | n > 0, 0 < m < n }, diseñar la gramática que lo genera.

Para diseñar la gramática estudiamos primero los casos para los valores mínimos para las variables m y n. Las restricciones obligan a que m sea estrictamente mayor que cero (valor mínimo 1), y n que sea estrictamente mayor que m, por lo que su valor mínimo será 2. Para sucesivos valores de m y n, se debe cumplir que el valor inferior de n esté acotado por m+1.

Para obtener la gramática conviene generar primero un conjunto de palabras, empezando por las más sencillas, para analizar el comportamiento.

Los valores básicos de n y m son 2 y 1 respectivamente. El caso base es aab. Sobre esta palabra podemos añadir más a, pero no más b. Tiene que haber siempre una a más que b. Esto lo podemos reformular como que partiendo del caso base (aab) podemos añadir aes libremente, pero cada vez que añadimos una b, debería ir emparejada con una a para evitar incumplir la restricción.

En forma de producciones podemos plantear:

S ::= aab | aS | aSb (caso base | añadir aes libres | añadir bs acompañadas de aes)

Obsérvese que el axioma acompaña a los terminales en una posición clave. No tendría sentido usar S ::= abS o S ::= Sab ya que eso generaría grupos de ab (abababab… en vez de aaaabbbb). También hay que analizar si valen tanto S ::= aS como S ::= Sa (en este caso sí, aunque recomendamos usar siempre la recursividad por la derecha).

Para facilitar la comprensión del problema, podemos representar el Lenguaje mediante una tabla. En un eje representamos cómo evolucionan las palabras al incrementar n, en el otro eje se representa para el superíndice m.

	n=1	n=2	n=3	n=4	n=5	n=6	n=7
m=1	-	aab	aaab	aaaab	aaaaab	aaaaaab	aaaaaaab
m=2	-	-	aaabb	aaaabb	aaaaabb	aaaaaabb	aaaaaaabb
m=3	-	-	-	aaaabbb	aaaaabbb	aaaaaabbb	aaaaaaabbb
m=4	-	-	-	-	aaaaabbbb	aaaaaabbbb	aaaaaaabbbb
m=5	-	-	-	-	-	aaaaaabbbbb	aaaaaaabbbbb
m=6	-	-	-	-	-	-	aaaaaaabbbbbb

La gramática debe poder generar cualquier palabra de la tabla aplicando las diferentes producciones en algún orden. Vemos que las palabras se incrementan en sentido horizontal añadiendo sucesivas aes, lo cuál se corresponde a aplicar la producción S ::= aS. Sin embargo, no es posible desplazarse mediante una producción concreta en sentido vertical incrementando libremente las bs, puesto que no podríamos evitar llegar a alguna casilla vacía que incumpla la restricción n > m. El origen de este problema está en que dicha restricción hace que las variables n y m no sean independientes.

La producción S ::= aSb representa un desplazamiento en diagonal descendente, que combinado con el desplazamiento horizontal, permite alcanzar cualquier celda de la tabla.

Aunque el problema planteado es lo suficientemente sencillo para resolverlo directamente, ahora planteamos otra forma de abordarlo, a través de una reformulación-descomposición y resolución por partes.

El problema lo podemos reformular a través de la restricción.

Donde teníamos $L = \{ a^n b^m \mid n > 0, 0 < m < n \}$, podemos transformar la restricción $0 < m < n$.

$\exists$ k | n = m + k > m ➔ k > 0 ➔ $L = \{ a^k a^m b^m \mid k, m > 0 \}$

	k=1	k=2	k=3	k=4	k=5
m=1	aab	aaab	aaaab	aaaaab	aaaaaab
m=2	aaabb	aaaabb	aaaaabb	aaaaaabb	aaaaaaabb
m=3	aaaabbb	aaaaabbb	aaaaaabbb	aaaaaaabbb	aaaaaaaabbb
m=4	aaaaabbbb	aaaaaabbbb	aaaaaaabbbb	aaaaaaaabbbb	aaaaaaaaabbbb
m=5	aaaaaabbbbb	aaaaaaabbbbb	aaaaaaaabbbbb	aaaaaaaaabbbbb	aaaaaaaaaabbbbb

Aquí vemos que es posible desplazarse en sentido horizontal añadiendo aes libremente, y en sentido vertical añadiendo parejas a-b, sin peligro de llegar a celdas prohibidas por las restricciones. Al representar los superíndices con m y k, hemos conseguido emplear variables independientes entre sí, cuando n y m no lo eran.

La reformulación nos permite descomponer el problema en dos partes diferenciadas. Vemos que hay dos estructuras independientes que son por un lado a^k y por otro lado $a^m b^m$.

Esto nos permite resolver el problema por partes:

```
a^k     a^m     b^m
|       |       |
|       ---+---
K          |
           M
```

Subgramática M:	**Subgramática K**:
Caso Base : ab	Caso Base : a
CB+Recursividad: aMb	CB+Recursividad: aK
λ Є L_M? no	λ Є L_K? no
M ::= ab | aMb	K ::= a | aK

Conexión de ambas subgramáticas. La estructura que combina M y K son bloques en paralelo, por lo que la unión se efectúa añadiendo S ::= KM :

S ::= KM
M ::= ab | aMb
K ::= a | aK

También se puede plantear $L = \{ a^m a^k b^m \mid k, m > 0 \}$ porque genera otra estructura válida, ésta vez de un bloque dentro de otro bloque.

```
a^m    a^k     b^m
|      |       |
|      K       |
|              |
------+------
      |
      M
```

Subgramática M:	**Subgramática K**:
Caso Base : ab	Caso Base : a
CB+Recursividad: aMb	CB+Recursividad: aK
λ Є L_M? no	λ Є Є L_K? no
M ::= ab | aMb	K ::= a | Ak

Conexión de ambas subgramáticas. En este caso, tenemos un bloque dentro de otro, así que el exterior tiene que ser capaz de derivar el interior. Si dejamos esta operación cómo última derivación de la gramática de M, necesitaremos sustituir las producciones que no contengan recursividad por otras que derivan el bloque K.

M ::= ab ➔ **M ::= aKb**
M ::= aMb ➔ M ::= aMb
K ::= a | aK

En este caso basta con sustituir M ::= ab por M ::= aKb

En caso de pertenecer lambda a L_M (el valor mínimo de m sería 0), tendríamos los siguientes cambios:

M ::= λ ➔ **M ::= K**
M ::= ab ➔ **M ::= aKb**
M ::= aMb ➔ M ::= aMb
K ::= a | aK