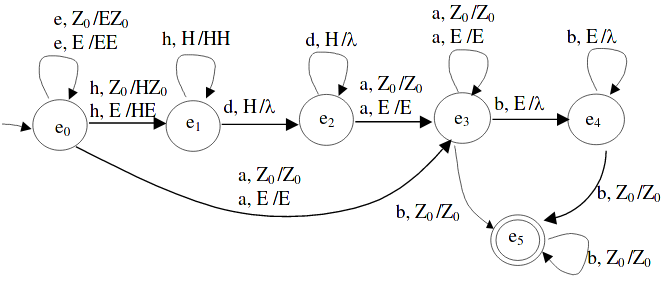
**Práctico 4 - Gramáticas libres de contexto**

**Ejercicio 1**

Describir el lenguaje aceptado por el siguiente autómata de pila:

APD=<{eo,e1,e2,e3,e4,e5},{e,h,d,a,b},{E,H, Z0}, δ ,e0, Z0, {e5}>



El lenguaje que acepta el APD es: {en hm dm aj bn+k / n, m >= 0; j, k >= 1}.

**Ejercicio 2**

Para cada uno de los siguientes lenguajes, definidos sobre el alfabeto:

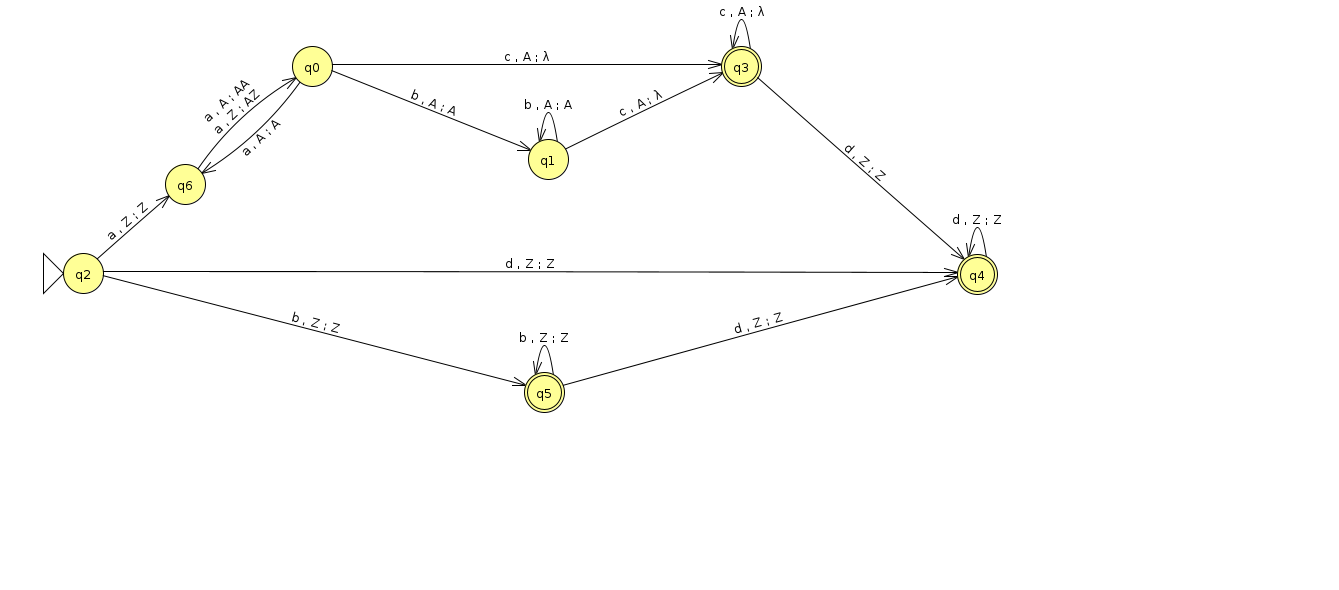
A = {a, b, c, d, e, h, x, y, z,0, 1, 2, 3, 4 }.

Diseñar y definir formalmente un autómata de pila que lo reconozca:

1. L1 = { a²k b2n ck dj / k, n, j ≥ 0 }.

Base: (aa)k (bb)n ck dj

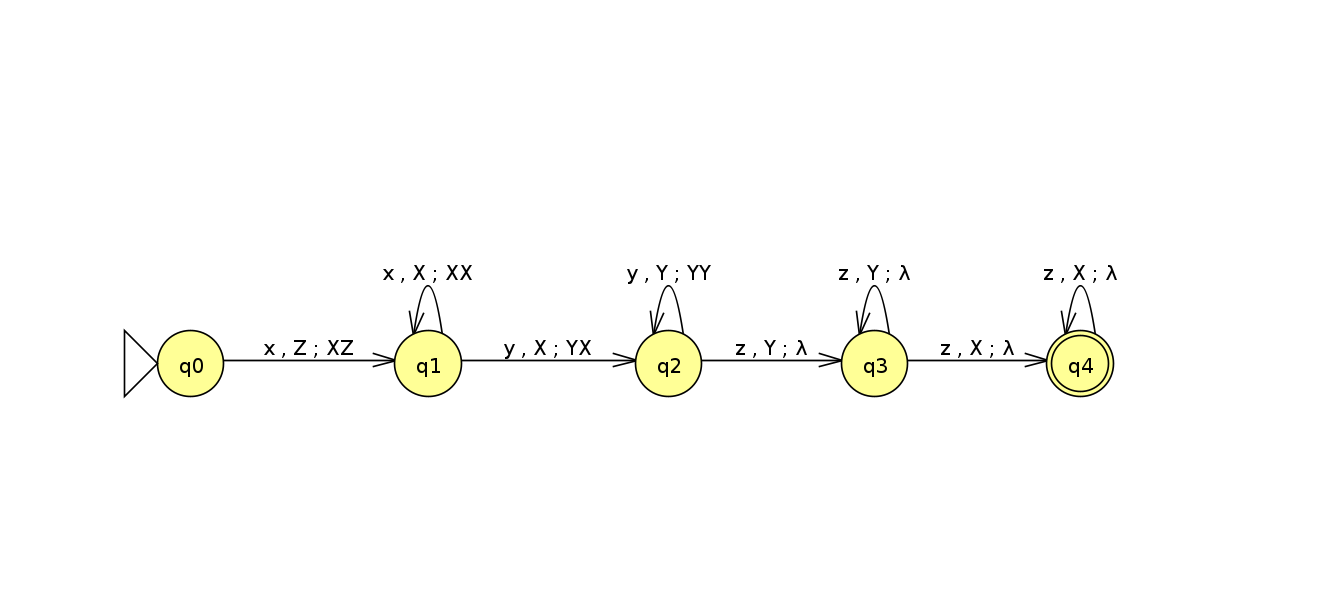
|  |  |  |  |
| --- | --- | --- | --- |
| **k** | **n** | **j** | **Cadena** |
| 0 | 0 | 0 | λ |
| 0 | 0 | 1 | d |
| 0 | 1 | 0 | bb |
| 0 | 1 | 1 | bbd |
| 1 | 0 | 0 | aac |
| 1 | 0 | 1 | aacd |
| 1 | 1 | 0 | aabbc |
| 1 | 1 | 1 | aabbcd |



1. L2 = { xr ys zt / t = r+s; r,s >= 1}.

Base = xr ys zs zr

Mínimo = xyzz.

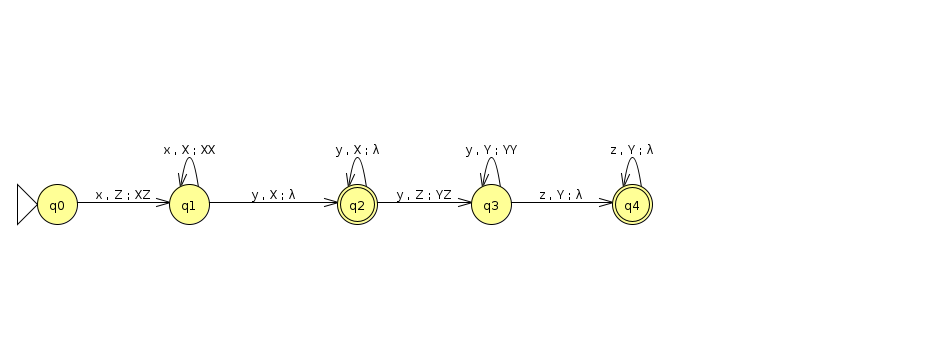


1. L3 = { xr ys zt / s = r+t; r,s >= 1}.

s = r + t <==> 1 = 1 + t <=> 1 - 1 = t <==> t = 0.

Base = xr yr yt zt

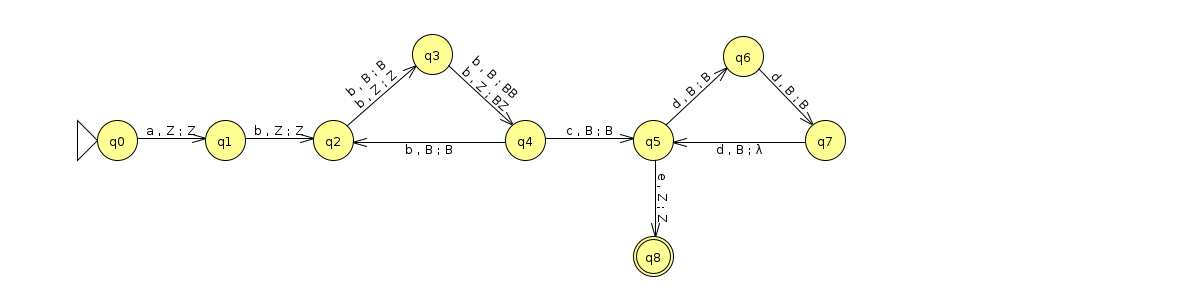
Minimo = xy.



1. L4 = {x / x = a Y e; donde Y = b3n c d3n, n >=1}.

Base = a (bbb)n c (ddd)n e

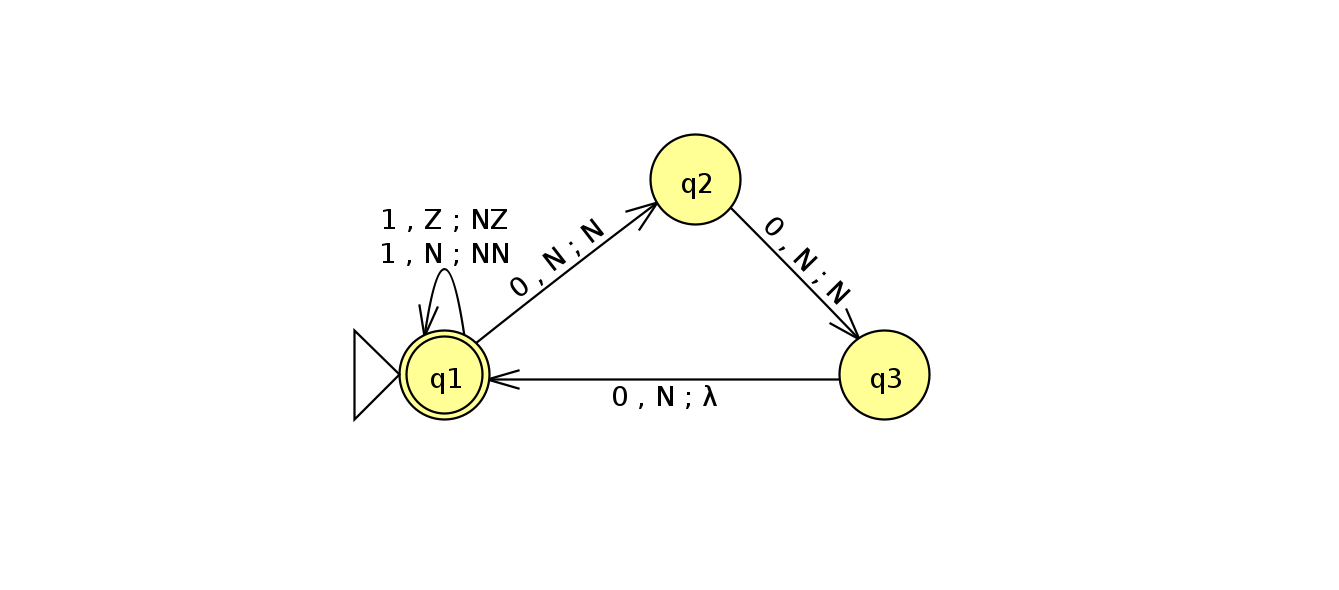
Mínimo = base.



1. L5 = {1n 0k / n >= 0, k = 3n}.

Base = 1n (000)n

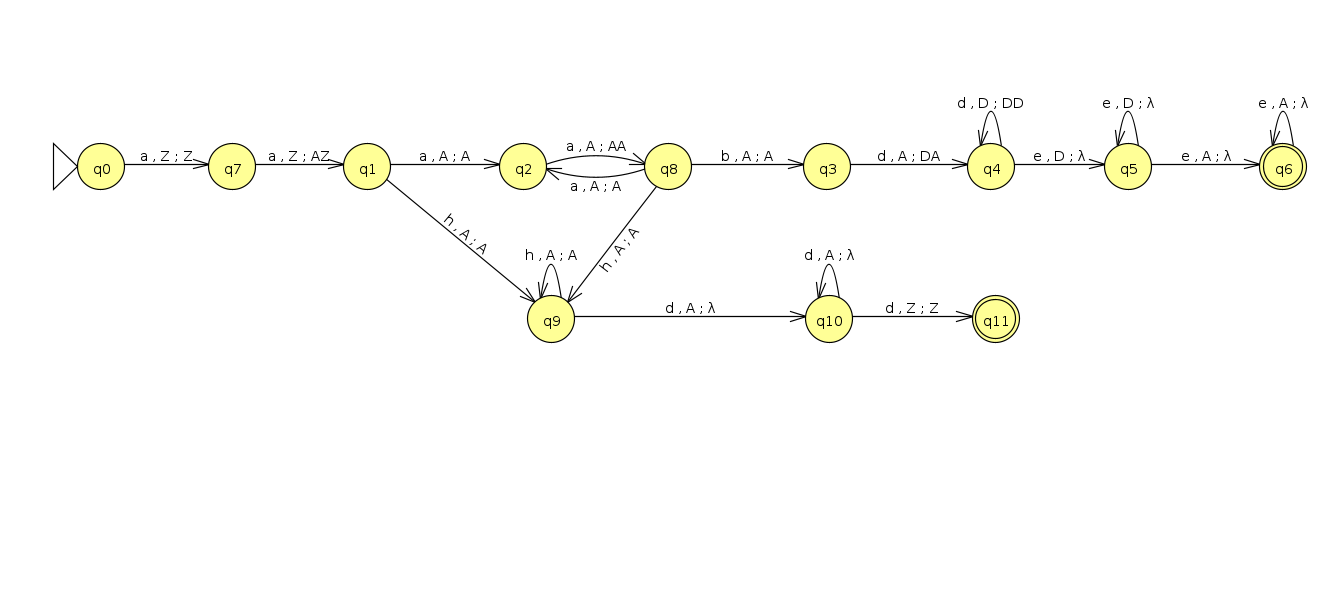
Mínimo = λ.



1. L6 = {a²n bi dk es+k / s, i, k > 0; n > s} U {a2k hj dk+1 / k, j > 0}.

Posibilidad 1: (aa)n (aa)s bi dk ek es / n, s, i, k >= 1.

Posibilidad 2: (aa)k hj dk d / k, j >= 1.

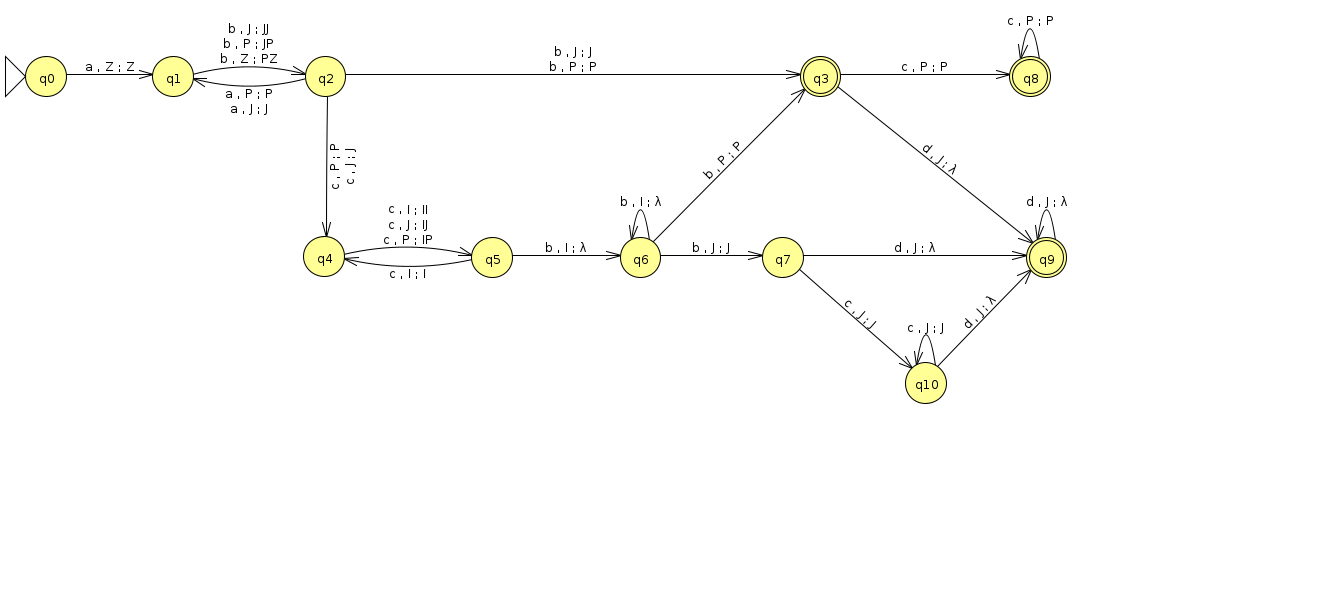


1. L7 = { (ab)j c2i bi+1 ck dn / i, j, k, n >= 0; j > n }.

Base = (ab)p (ab)j (cc)i bi b ck dj / p >= 1; j, i, k >= 0.

|  |  |  |  |
| --- | --- | --- | --- |
| **j** | **i** | **k** | **Cadena** |
| 0 | 0 | 0 | abb |
| 0 | 0 | 1 | abbc |
| 0 | 1 | 0 | abccbb |
| 0 | 1 | 1 | abccbbc |
| 1 | 0 | 0 | ababbd |
| 1 | 0 | 1 | ababbcd |
| 1 | 1 | 0 | ababccbbd |
| 1 | 1 | 1 | ababccbbcd |

Pruebas por error: ababbdd, ababbddd



**Ejercicio 3**

Para el siguiente APD llamado A, definir el lenguaje generado por comprensión.

A = <Q = {0, 1} , ∑= {a, b, c} , Г = { z0, A, B, C} , δ, 0 , z0, F = {0}> donde:

δ = (0, a, z0) = (1, Az0)

δ = (0, b, z0) = (1, Bz0).

δ = (1, a, A) = (1, AC).

δ = (1, c, A) = (1, λ)

δ = (1, b, B) = (1, BC).

δ = (1, c, B) = (1, λ).

δ = (1, c, C) = (1, λ).

δ = (1, λ, z0) = (0, z0).

Estado 0:

Si recibe una a y había z0, apila A y va al estado 1.

Si recibe una b y había z0, apila B y va al estado 1.

Estado 1:

Si recibe una a y había una A, apila AC y se mantiene en 1 (cuenta aes usando la C)

Si recibe una b y había una B, apila BC y se mantiene en 1 (cuentas bes usando la C).

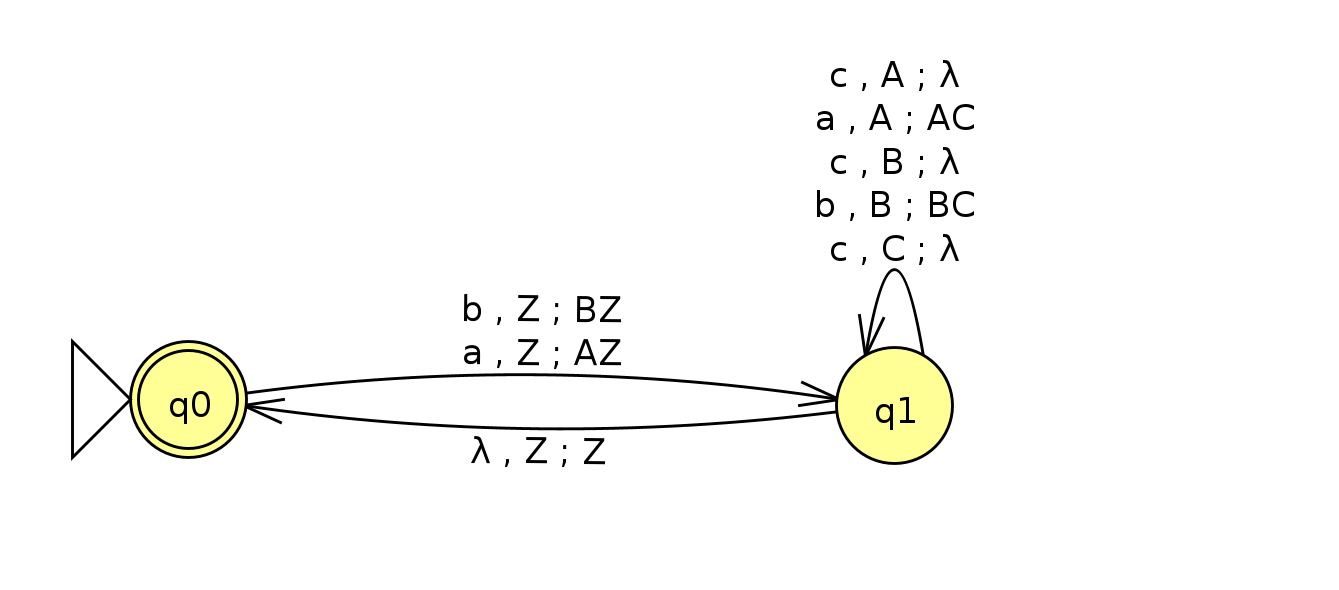
Si recibe una c y había una A, desapila y queda C en el tope, se mantiene en 1.

Si recibe una c y había una B, desapila y queda C en el tope, se mantiene en 1.

Si recibe una c y había una C, desapila y se mantiene en 1.

Si recibe una c y había una B, desapila y se mantiene en 1.

Si recibe λ y había z0, vuelve a estado 0 y finaliza.

El APD A genera L = {aj cj bk ck / j, k >= 0}.