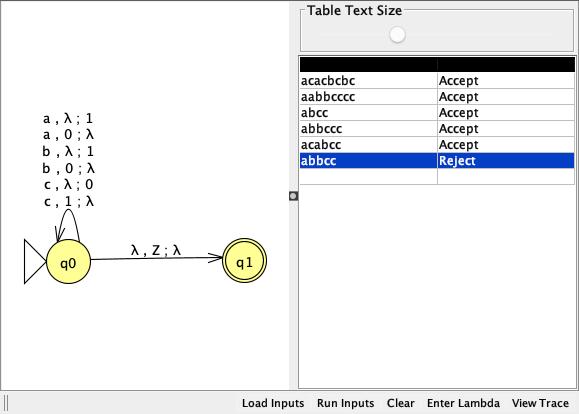
CSci 435: Formal Languages and Automata

Instructor: Dr. M. E. Kim Name: \_\_\_\_\_Derek Trom\_\_\_\_\_\_\_

**Home Assignment 5: 103/100 points + 15 points (optional)**

In any (N/D)PDA, assume that a start stack symbol z is already in the stack; so, you don’t have to insert z into the stack at the beginning of transition.

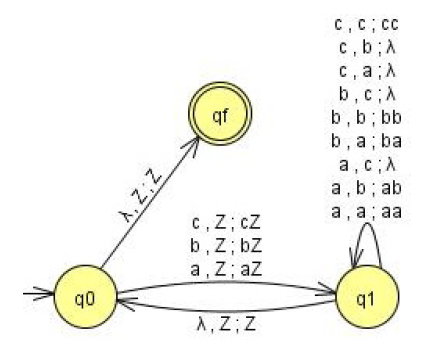
Q1. [26/20] For a given language L = { *w* | *na*(*w*) + *nb*(*w*) = *nc*(*w*) } where Σ = Γ = {*a*, *b, c*}

1. [6/10] Construct a PDA M that accepts L with Σ = Γ = {*a*, *b, c*}
   1. 

The only time we will consume a symbol on the stack is either

1. the input = *a* or b, and the top of the stack = c, or
2. the input = c and the top of the stack = *a* or b.

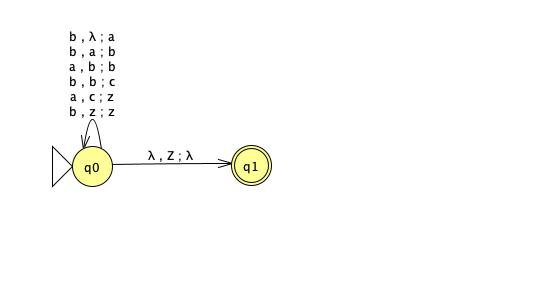
In all the other cases, we simply put the same symbol as input to the stack.

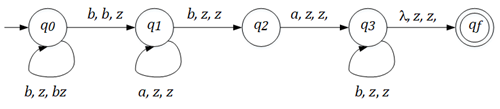


1. [10/10] Show the sequence of instantaneous descriptions for the acceptance of *acacbcbc* by M in 1).
   1. (q0, acacbcbc, z)
   2. (q0, cacbcbc, 1z)
   3. (q0, acbcbc, z)
   4. (q0, cbcbc, 1z)
   5. (q0, bcbc, z)
   6. (q0, cbc, 1z)
   7. (q0, bc, z)
   8. (q0, c, 1z)
   9. (q0, λ, z)
   10. (q1, z)
2. [10/10, optional] Give a CFG G that generates L, L(G) = L.
   1. S → SS | aSc | bSc | cSa | cSb | λ

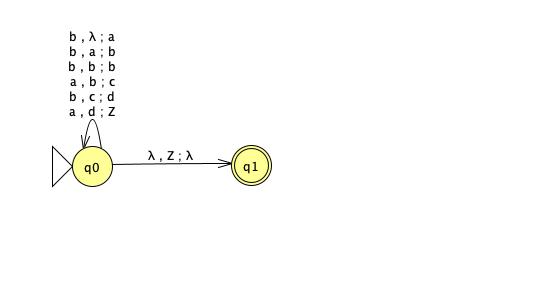
Q2. [20/20] Construct an NPDA for the following languages.

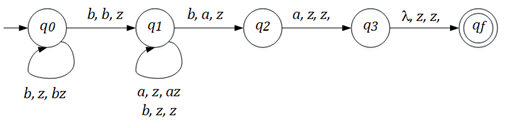
1. [8/10] L1 = {*bba*\**bab*\* }



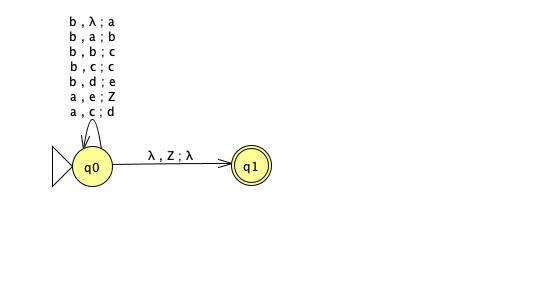


1. [8/10] L2 = {*bbb\*aba* }

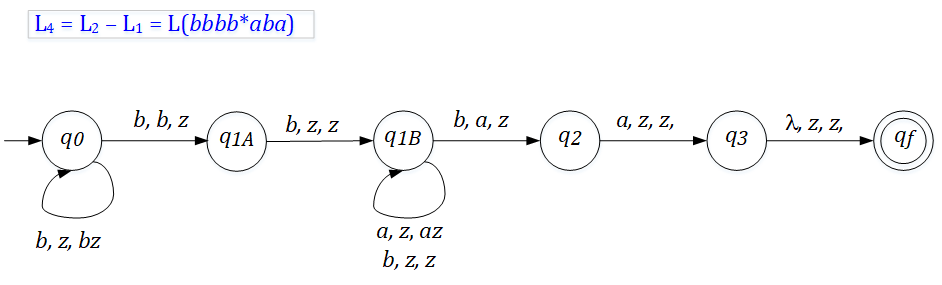




1. [4/5, optional] L4 = L2 – L1.



L4 = L2 – L1 = L(*bbbb*\**aba*) since any w ∈L2 must have a suffix *aba* while the only string with suffix *aba* in L1 is *bbaba*.



Q3. [10/10] Give the language that is accepted by the NPDA M in a formal expression (including a regular expression) where M = ({*q0, q1, q2*}, {*a, b*}, {*a, b*, z}, δ, *q0*, z, { *q0* , *q1*, *q2*}), with transitions

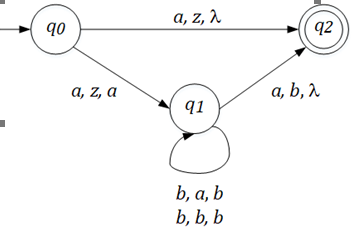
♦ δ(*q0*, *a*, z) = {(*q1*, *a*), (*q2*, λ)},

♦ δ(*q1*, *b*, *a*) = {(*q1*, *b*)},

♦ δ(*q1*, *b*, *b*) = {(*q1*, *b*)},

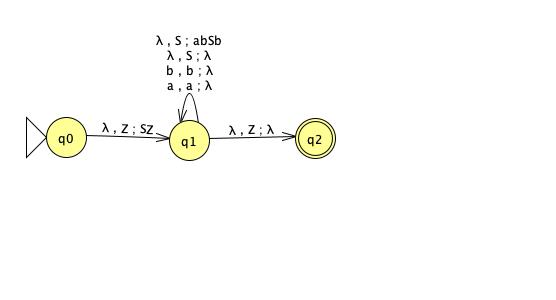
♦ δ(*q1*, *a*, *b*) = {(*q2*, λ)},

1. L = {a + abb\*a} , L = { a } ⋃ { abna | n > 0 }

 L = {**λ**, *a*} ∪ L(*a*bb\**a*)

Q4. [17/20] (A) Construct a NPDA that accepts the language defined by the given grammar and (B) give the language in a formal expression (including a regular expression).

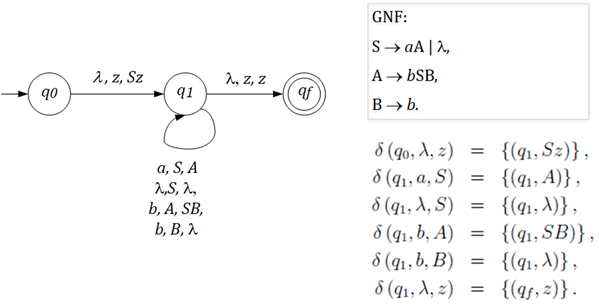
1. 9/10 S → *ab*S*b* | λ.

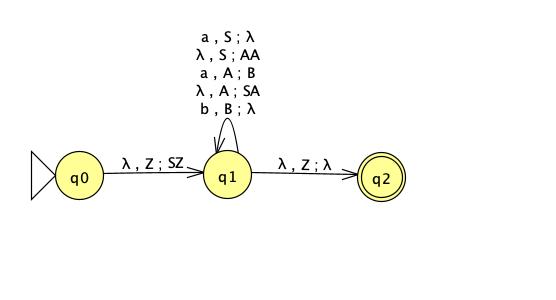


* 1. L = { (ab)nbn | n ≥ 0 }

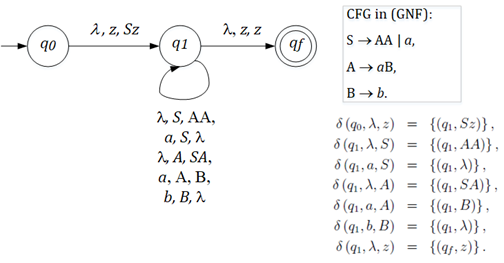
In GNF: S → *a*A | λ, A → *b*SB, B → *b*.

Then, a PDA is:



1. 8/10 S → AA | *a*, A → SA | *ab*
   1. S→AA|a, A→AAA|aA|aB, B→b
   2. Into GNF:
   3. S→aACA|aBCA|aAA|aBA|a, A→aAC|aBC|aA|aB, B→b
   4. C→aACAC|aACBC|aACA|aBCA|aAA|aAB|aBA
   5. 
   6. L={ a + (abab + aabab + abaab + aabaab)\* }

Convert A → *ab* intoA → *aB* and  *B* → *b* in GNF. Then,



Hint: Convert the grammar into Greibach Normal Form, then apply Thm. 7.1.

Q5. [20/20] Find a (minimal) Context-Free Grammar that generates the language accepted by the NPDA M where M = ({*q0, q1*}, {*a, b*}, {*A*, z}, δ, *q0*, z, {*q1*}), with the transitions

♦ δ(*q0*, *a*, z) = {(*q0*, *Az*)},

♦ δ(*q0*, *b*, *A*) = {(*q0*, *AA*)},

♦ δ(*q0*, *a*, *A*) = (*q1*, λ).

Simplify the production rules by eliminating the useless variables and productions.

Transitions:

δ(q0, a, z) = {(q0, Az)}

δ(q0, b, A) = {(q0, AA)}

δ(q0, a, A) = {(q1, λ)}

δ(q1, λ, A) = {(q1, λ)}

δ(q1, λ, z) = {(q2, λ)}

Productions:

(q0zq0) → a(q0Aq0)(q0zq0) | a(q0Aq1)(q1zq0) | a(q0Aq2)(q2zq0)

(q0zq1) → a(q0Aq0)(q0zq1) | a(q0Aq1)(q1zq1) | a(q0Aq2)(q2zq1)

(q0zq2) → a(q0Aq0)(q0zq2) | a(q0Aq1)(q1zq2) | a(q0Aq2)(q2zq2)

(q0Aq0) → b(q0Aq0)(q0Aq0) | b(q0Aq1)(q1Aq0) | b(q0Aq2)(q2Aq0)

(q0Aq1) → b(q0Aq0)(q0Aq1) | b(q0Aq1)(q1Aq1) | b(q0Aq2)(q2Aq1)

(q0Aq2) → b(q0Aq0)(q0Aq2) | b(q0Aq1)(q1Aq2) | b(q0Aq2)(q2Aq2)

(q0Aq1) → a

(q1Aq1) → λ

(q1zq2) → λ

Convert to variables:

A → aDA

B → aDB

C → aDC | aEY | aXZ

D → bDD

E → bDE | bEY | bXY

F → bDF

X → a

Y → λ

Z → λ

Remove useless:

C →aEY | aXZ

E → bEY | bXY

X → a

Y → λ

Z → λ

Remove lambas:

S → aE | aX

E → bE | bX

X → a

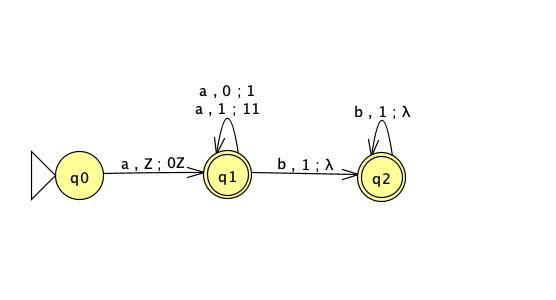
Final after removing X:

S → aE | aa

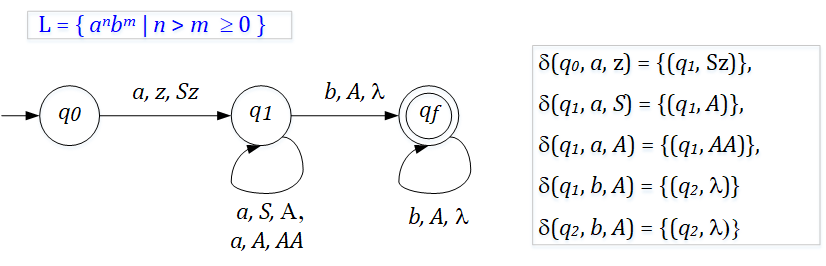
E → bE | ba

Q6. [10/10] Construct a Deterministic-PDA that accepts L= { *anbm* | 0 ≤ *m* < *n* } to show L is a Deterministic-CFL.

M<N, M>=0



This DPDA puts a token on the stack when input is an *a* (except for the first one), then consumes one when the input is a *b*. It goes into a dead configuration when input b has consumed all tokens in the stack.

  
 S → AS | *a*S*b*

A → *a*A | λ