CSci 435: Formal Languages and Automata

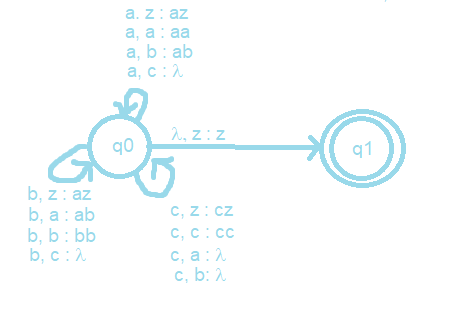
Instructor: Dr. M. E. Kim Name: Sam Dressler

**Home Assignment 5: 88/100 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. [27/20] For a given language L = { *w* | *na*(*w*) + *nb*(*w*) = *nc*(*w*) } where Σ = Γ = {*a*, *b, c*}

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



The cycles from q0 can be compressed to one cycle but were split here for ease of reading.

1. [10/10] Show the sequence of instantaneous descriptions for the acceptance of *acacbcbc* by M in 1).

(q0, acacbcbc, z)

˫ (q0, cacbcbc, az) //’a’ as input symbol and ‘z’ on top of stack so push ‘az’

˫ (q0, acbcbc, z) // ‘c’ as input symbol and ‘a’ on top of stack so pop ‘a’

˫ (q0, cbcbc, az) // ‘a’ as input symbol and z’ on top of stack so push ‘az’

˫ (q0, bcbc, z) // ‘c’ as input symbol and ‘a’ on top of stack, so pop ‘a’

˫ (q0, cbc, bz) // ‘b’ as input symbol and ‘z’ on top of stack, so push ‘bz’

˫ (q0, bc, z) // ‘c’ as input symbol and ‘a’ on top of stack, so pop ‘b’

˫ (q0, c, bz) // ‘b’ as input symbol and ‘z’ on top of stack, so push ‘bz’

˫ (q0, λ, z) // ‘c’ as input symbol and ‘a’ on top of stack, so pop ‘b’

˫ (q1, λ, z) // ‘λ’ as input symbol and ‘z’ on top of stack, so we are finished.

1. [7/10, optional] Give a CFG G that generates L, L(G) = L.

S-> aX | bX | λ

X -> cS | Sc

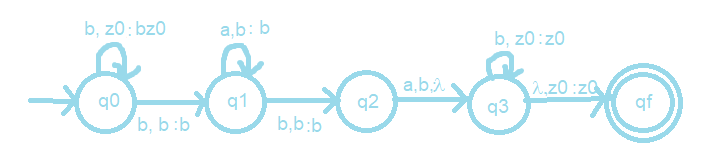
For L = { *anbmck* | *k = n+m* }, CFG is: S → *a*Sc | B, B → bBc | λ

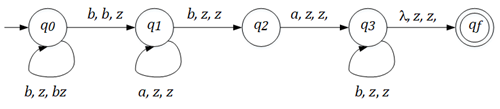
For L = { *w* | *na*(*w*) + *nb*(*w*) = *nc*(*w*) }, the orders of *a, b, c* are irrelevant.

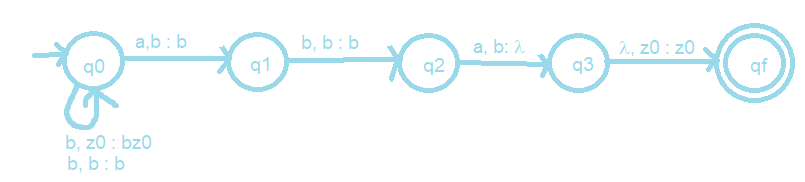
So, its CFG G is:

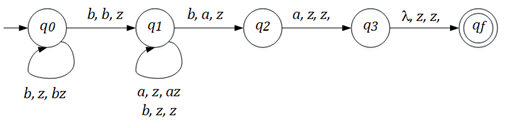
S → *a*Sc | bSc | *c*S*a* | cSb | SS | λ.

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

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



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



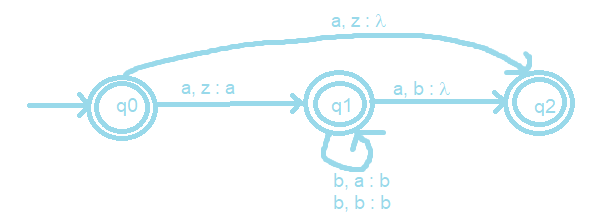
1. [0/5, optional] L4 = L2 – L1.

\*Q3. [7/10] 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

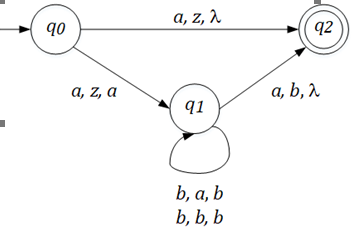
♦ δ(*q0*, *a*, z) = {(*q1*, *a*), (*q2*, λ)},

♦ δ(*q1*, *b*, *a*) = {(*q1*, *b*)},

♦ δ(*q1*, *b*, *b*) = {(*q1*, *b*)},

 ♦ δ(*q1*, *a*, *b*) = {(*q2*, λ)},

L(M) = {l + a + ab\* + ab\*a}

 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).

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

1. 10/10 S → *ab*S*b* | λ.

Greibach Normal Form:

S->aBSB | aBB

B -> b

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

NPDA for above GNF of Grammar:

M = ({q0, q1, q2}, {a, b, l), {B, S, z}, d, q0, z, {q2})

d{

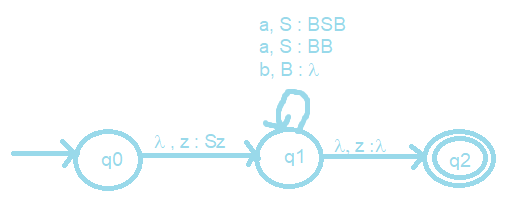
d (q­0, l, z) = {(q1, Sz)}

d (q­1, a, S) = {(q1, BSB), (q1, BB)}

d (q­0, b , B) = {(q1, l)}

d(q1, l, z) = {(q2, z)}

}



L = {(ab)nbn | n > 0}

Regex: abb + (ab)\*b\*

1. 7/10 S → AA | *a*

A → SA | *ab*.

M = ({*q0, q1*}, {*a, b*}, {*A*, z}, δ, *q0*, z, {*q1*}), with the transitions

S->AA | a

A->SA | ab

S = A1, A = A2

GNF:

A1 -> aA2A2 | aBA2 | aA2B2A2 | aBB2A2 | a

A2 -> aA2 | aB | aA2B2 | aBB2

B2 -> aA2A2A2 | aBA2A2 | aA2B2A2A2 | aBB2A2A2

B2 -> aA2A2A2B2 | aBA2A2A2B2 | aA2B2A2A2A2B2 |aBB2A2A2A2B2

B -> b

NPDA for above GNF of Grammar:

M = ({q0, q1, q2}, {a, b, l), {A1, A2, B, B2, z}, d, q0, z, {q2})

d{

d(q0, l, z) = {(q1, A1z)}

d(q1, a, A1) = {(q1, A2A2), (q­1, BA2), (q1, A2B2A2), (q1, BB2A2), (q1, l)}

d(q1, a, A2) = {(q1, l), (q1, B), (q1, A2B2), (q1, BB2)}

d(q1, a, B2) = {(q1, A2A2A2), (q1, BA2A2), (q1, A2B2A2A2), (q1, BB2A2A2),

(q1, A2A2A2B2), (q1, BA2A2A2B2), (q1, A2B2A2A2A2B2),

(q1, BB2A2A2A2B2)}

d(q1, b, B) = {(q1, l)}

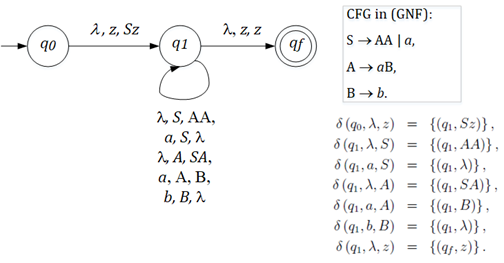
d(q1, l, z) = {(q2, z)}

}

L(G) = (aab + aa + aba+abab)(a, ab)n | n >= 0

Regex = a(a,ab)(a,ab)\* + ab(a,ab)(a,ab)\*

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



Q5. [7/20] 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.

S-> (q0zq­­0) | (q­­0zq1)

(q0zq0) -> a(q0Aq1) (q1zq0) | a(q0Aq0) (q0zq0)

(q0zq1) -> a(q0Aq0) (q0zq1) | a(q0Aq1) (q1zq1)

(q0Aq0) -> b(q0Aq0) (q0Aq0) | b(q0Aq1) (q1Aq0)

(q0Aq1) -> b(q0Aq0) (q0Aq1) | b(q0Aq1) (q1Aq1)

(q0Aq1)-> a

There are no transitions for (q1Aq1), (q1Aq0), and (q1zq1), so they are useless and can be removed.

S-> (q0zq­­0) | (q­­0zq1)

(q0zq0) -> a(q0Aq1) (q1zq0) | a(q0Aq0) (q0zq0)

(q0zq1) -> a(q0Aq0) (q0zq1)

(q0Aq0) -> b(q0Aq0) (q0Aq0)

(q0Aq1) -> b(q0Aq0) (q0Aq1)

(q0Aq1)-> a

L(M) = L(*ab*\**a*). The CFG for L(M) is: S → *a*B*a*, A → *b*B | λ.

If we construct NPDA by Thm. 7.2, similar to Example 7.8:

The given transitions doesn’t satisfy the assumption (1)

s.t. PDA has a single final state that is entered iff the stack is empty.

So, let’s create a new final state *q2*, making the stack empty:

**δ(*q1,*, λ*,* A) = {( *q1, λ*)}(4). δ(*q1,*, λ*,* z) = {( *q2,* λ)}(5).**

So, the Start variable: **(*q0*z*q2*)**

δ(*q0,*, *a, z*) = {( *q0, Az*)} (1),

δ(*q0,*, *b,* A) = {( *q0, AA*)} (2),

δ(*q0,*, *a,* A) = {( *q1, λ*)}(3) 🡺 **(*q0*A*q1*)(3)→ *a***

δ(*q1,*, λ*,* A) = {( *q1, λ*)}(4)  🡺 **(*q1*A*q1*)(4)→ λ**

δ(*q1,*, λ*,* z) = {( *q2,* λ)}(5) 🡺 **(*q1*z*q2*)(5)→ λ**

Reminder: δ(*qi, a, A*) =(*qj ,BC*)**eq.7.6**) generates (*qiAqk*) → *a*(*qjBql*)(*qlCqk*).

From the transitions (1) **δ(*q0,a,z*)={(*q0, Az*)}(1),** we get the set of productions in G

**(*q0*z*q0*) → *a*(*q0*A*q0*)(*q0*z*q0*)**|*a*(*q0*A*q1*)(*q1*z*q0*) |*a*(*q0*A*q2*)(*q2*z*q0*)

**(*q0*z*q1*) → *a*(*q0*A*q0*)(*q0*z*q1*)**|*a*(*q0*A*q1*)(*q1*z*q1*) |*a*(*q0*A*q2*)(*q2*z*q1*)

**(*q0*z*q2*) → *a*(*q0*A*q0*)(*q0*z*q2*)**|*a*(*q0*A*q1*)(*q1*z*q2*) |*a*(*q0*A*q2*)(*q2*z*q2*)

So, From the transitions (2) **δ(*q0,b,A*)={(*q0, AA*)}(2),** we get the set of productions in G

**(*q0* A *q0*) → *b*(*q0*A*q0*)(*q0*A*q0*)** | *b*(*q0*A*q1*)(*q1*A*q0*) | *b*(*q0*A*q2*)(*q2*A*q0*)

**(*q0* A *q1*) → *b*(*q0*A*q0*)(*q0*A*q1*)** | *b*(*q0*A*q1*)(*q1*A*q1*) | *b*(*q0*A*q2*)(*q2*A*q1*)

**(*q0* A *q2*) → *b*(*q0*A*q0*)(*q0*A*q2*)** | *b*(*q0*A*q1*)(*q1*A*q2*) | *b*(*q0*A*q2*)(*q2*A*q2*)

The useless variable that doesn’t occur on the left side of any production:

(*q1*z*q0*), (*q1*z*q1*), (*q2*z*q0*), (*q2*z*q1*), (*q2*z*q2*), (*q1*A*q0*), (*q1*A*q2*), (*q2*A*q0*), (*q2*A*q1*), (*q2*A*q2*)

So, the corresponding rules are removed.

Since **(*q0*A*q1*)(3)→ *a,* (*q1*A*q1*)(4)→ λ** and **(*q1*z*q2*)(5)→ λ** are the rules that yield a terminal symbol, any rule which doesn’t derive these variables on the sentinel form doesn’t yield a terminal string, so it’s useless, either.

For a convenience, let’s rename the variables.

Start variable: S = **(*q0*z*q2*)**

**A = (*q0*z*q0*), B =(*q0*z*q1*), C=(*q1*z*q2*),**

**P =(*q0*A*q0*), Q = (*q0* A *q1*), X = (*q0* A *q2*) Y= (*q1*A*q1*)**

**(*q0*A*q1*)(3)→ *a***: **Q → *a***

**(*q1*A*q1*)(4)→ λ: Y → λ**

**(*q1*z*q2*)(5)→ λ: C → λ**

**(*q0*z*q0*) → *a*(*q0*A*q0*)(*q0*z*q0*)** : A → *a*PA - useless

**(*q0*z*q1*) → *a*(*q0*A*q0*)(*q0*z*q1*)** : B → *a*PB - useless

**(*q0*z*q2*) → *a*(*q0*A*q0*)(*q0*z*q2*)**|*a*(*q0*A*q1*)(*q1*z*q2*) **S** → *a*PS |***a*QC**

**(*q0* A *q0*) → *b*(*q0*A*q0*)(*q0*A*q0*)** : P → *b*PP -- useless

**(*q0* A *q1*) → *b*(*q0*A*q0*)(*q0*A*q1*)** | *b*(*q0*A*q1*)(*q1*A*q1*):**Q** → *b*PQ | ***a*QY**

**(*q0* A *q2*) → *b*(*q0*A*q0*)(*q0*A*q2*)** X → *b*PX -- useless

**(*q0*z*q0*), (*q0*z*q1*), (*q0* A *q0*)** and **(*q0* A *q2*)** are also the useless variables since they never derive the terminal symbols. So, these variables and the production rules that involve them are also eliminated.

Thus, the final production rules are:

**(*q0*A*q1*)(3)→ *a*** | *b*(*q0*A*q1*)(*q1*A*q1*): **Q → *a* | *b*QY**

**(*q1*A*q1*)(4)→ λ: Y → λ**

**(*q1*z*q2*)(5)→ λ: C → λ**

**(*q0*z*q2*) →** *a*(*q0*A*q1*)(*q1*z*q2*) **S** → ***a*QC**

**~~(~~*~~q~~~~0~~* ~~A~~ *~~q~~~~1~~*~~) →~~** *~~b~~*~~(~~*~~q~~~~0~~*~~A~~*~~q~~~~1~~*~~)(~~*~~q~~~~1~~*~~A~~*~~q~~~~1~~*~~):~~ **~~Q~~** ~~→~~***~~b~~*~~QY~~**

If we further simplify them, S → *a*Q, Q → *a* | bQ

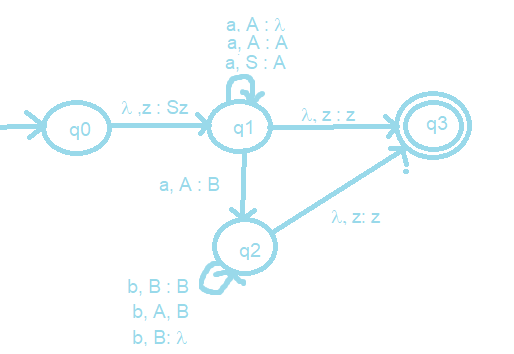
i.e. S → *a*Q*a*, Q → *b*Q | λ.

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

Grammar for L

S -> aA | a

A -> aA | aB | a

B -> bB | b