Exercise 9.1

Define a PDA (push-down automaton) that accepts the language of all correctly nested sequences of round or square brackets.

• a sequence consists of any number of bracket pairs.

• Each bracket pair begins with an opening bracket [or (and ends with the corresponding closing bracket] or). Between opening and closing bracket there can be any sequence of (possibly nested) bracket pairs.

Here are some examples of correctly nested sequences:

[]
()[]()
(())[[]]
[()]([() []])()

Also the empty string ε is allowed. In contrast, the following strings should not be accepted.

[[])	opening square bracket [closed by round bracket)
([]	closing bracket) is missing
) [] (	has to begin with an opening bracket

S → (S) | [S] | SS | ε

(1) z0, k0 z1, k0 S push start symbol S on stack
(2.1) z1, (→ (z1, ε consume matching symbol (
(2.2) z1,) →)z1, ε consume matching symbol)
(2.3) z1, [→ [z1, ε consume matching symbol [
(2.4) z1,] →]z1, ε consume matching symbol]
(3.1) z1, S → εz1, (S) expand with production S → (S)
(3.2) z1, S → εz1, [S] expand with production S → [S]
(3.3) z1, S → εz1, SS expand with production S → SS
(3.5) z1, S → εz1, ε expand with production S → ε
(4) z1, k0 → εze, k0 switch to end state when stack empty

Exercise 9.2

a) Define a PDA that accepts the language of the following context-free grammar:

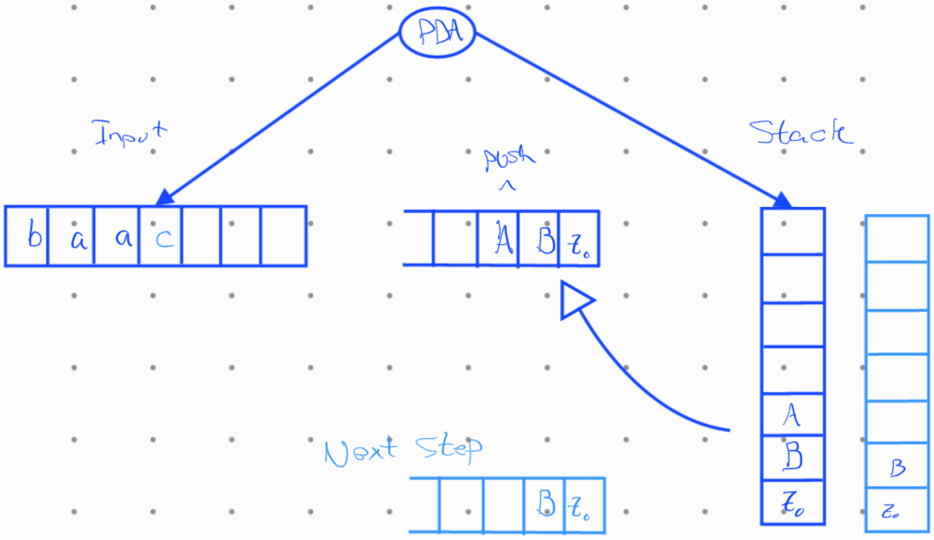
S → bA
A → BAB | c
B → aa | b

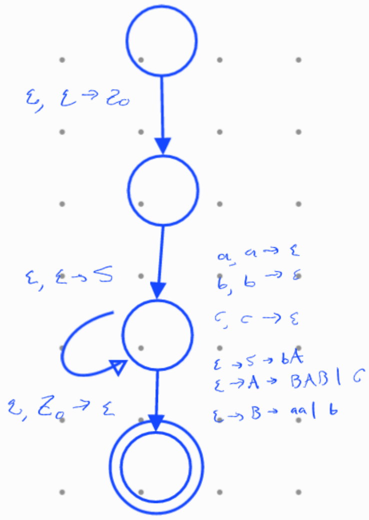
M = ({q},{a,b,c},{A,B,S},δ,q,S,θ)
δ =(q,b,S) = {(q,A)}
δ =(q,c,A) = {(q,BAB), (q,ε)}
δ =(q,aa,B) = {(q,ε)}
δ =(q,b,B) = {(q,ε)}

b) Show how the PDA accepts **baacb**. (Tip: It might help to draw a derivation tree first).

(Left sentential form)Left Most derivation: S,bA, bBAB, baaAB, baacB, baacb

- Top stack is matched to a rule, and then popped
- Push right hand side of production onto stack
- Next step until z0





Exercise 9.3 - obligatory (6 points)

Let the following context-free grammar be given:

$S \rightarrow aAc$

$A \rightarrow BA \mid c$

$B \rightarrow ab \mid \varepsilon$

a) Define a PDA that accepts the language of the grammar.

M = ({q},{a,b,c},{A,B,S},δ,q,S,θ)

δ =(q,(a,c),S) = {(q,A,q)}

δ =(q,c,A) = {(q,BA), (q,ε)}

δ =(q,ab,B) = {(q,ε)}

(1) z0, k0 z1, k0 S push start symbol S on stack
(2.1) z1, a → az1, ε consume matching symbol a
(2.2) z1, b → bz1, ε consume matching symbol b
(2.3) z1, c → cz1, ε consume matching symbol c
(3.1) z1, S → εz1, aAc expand with production S → aAc
(3.2) z1, A → εz1, BA expand with production A → BA
(3.3) z1, A → εz1, c expand with production A → c
(3.4) z1, B → εz1, ab expand with production B → ab
(3.5) z1, B → εz1, ε expand with production B → ε
(4) z1, k0 → εze, k0 switch to end state when stack empty

b) Give a sequence of configuration steps that shows that the PDA accepts the string **aabcc**.
Tip: First think about what a derivation tree for the string looks like

state stack input.	next operation
(z0, k0, aabcc)	(1) push start symbol S on stack
→ (z1, k0S, aabcc)	(3.1) expand with S → aAc
→ (z1, k0cAa, aabcc)	(2.1) consume symbol a
→ (z1, k0cA, abcc)	(3.2) expand with A → BA
→ (z1, k0cAB, abcc)	(3.4) expand with B → ab
→ (z1, k0cbaA, abcc)	(3.3) expand with A → c
→ (z1, k0cbac, abcc)	(2.3) consume symbol c
→ (z1, k0cba, abc)	(2.1) consume symbol a
→ (z1, k0cb, bc)	(2.2) consume symbol b
→ (z1, k0c, c)	(2.3) consume symbol c
→ (z1, k0, e)	(4) end state reached, input also empty
→ (ze, k0, e)	

aAc
aBAc
aabcc

S → aAc
x aAc → BAc
BAc → abAc
x abAc → Ac
Ac → cc
x cc → ε

expand aAc
consume a
expand ab

Exercise 9.4

Grammar G with start symbols S is defined as following:

S → aA | Bd
A → BC | a
B → bBa | C
C → cCb | ε

Compute for this grammar the properties nullable, First, and Follow.

A → ε. set nullable(A) = true
B → ε. set nullable(B) = true
C → ε. set nullable(C) = true

	Nullable	First	Follow
S	-	a, d, b, c	$\$$
A	true	a, b, c, ε	$\$$
B	true	b, c, ε	c, a, d
C	true	c, ε	b, $\$$, c, a, d

First(S) = {a} u
(S) εd = u {d}
(S) Bd = u {b} u {c}
First(A) = {a,b,c,ε}
First(B) = {b,c,ε}
First(C) = {c,ε}
Follow(S) = {$}
Follow(A) = follow(S)
Follow(B) = {a}
(B) Bd = u {d}
(B) C = u {c}
Follow(C) = follow(A) u {b}

B →
αAβ = follow(A) = first (β)
αA = follow(A) = follow (β)

Exercise 9.5 - obligatory (14 points)

Let G be the following context-free grammar with start symbols S:

S → BSA | aAB
A → bA | ε
B → cBAa | A

A → ε. set nullable(A) = true
B → ε. set nullable(B) = true

a) Compute for all non-terminal symbols of the grammar the properties:

(1) nullable(X)
(2) First(X)

	Nullable	First	Follow
S	-	a, b, c, ε	$, b
A	true	b, ε	c, b, a, $
B	true	c, b, ε	a, b, c, $

(3) Follow(X).

First(S) = {a,c,b,ε}
First(A) = {b,ε}
First(B) = {c,b,ε}
Follow(S) = {$} u {b}
Follow(A) = Follow(A) u {c,b}
Follow(B) = {b}

B →
αAβ = follow(A) = first (β)
αA = follow(A) = follow (B)

b) Compute then:

(1) First(cBA) = {c}

(2) First(ASB) = {a,b,c}

(3) First(ABa) = {b,c,a}