Seminar 13

I. PDA

1. Find a PDA that accepts the language $L = \{ww^R, w \in \{a,b\}^+\}$ // Alexandra T.

$M = (\{q_0, q_1, q_f\}, \{a,b\}, \{Z_0, A, B\}, \delta, q_0, Z_0, \{q_f\})$

$\delta(q_0, a, Z_0) = \{(q_0, AZ_0)\}$

$\delta(q_0, b, Z_0) = \{(q_0, BZ_0)\}$

$\delta(q_0, a, A) = \{(q_0, AA), (q_1, \epsilon)\}$

$\delta(q_0, a, B) = \{(q_0, AB)\}$

$\delta(q_0, b, A) = \{(q_0, BA)\}$

$\delta(q_0, b, B) = \{(q_0, BB), (q_1, \epsilon)$

$\delta(q_1, a, A) = \{(q_1, \epsilon)\}$

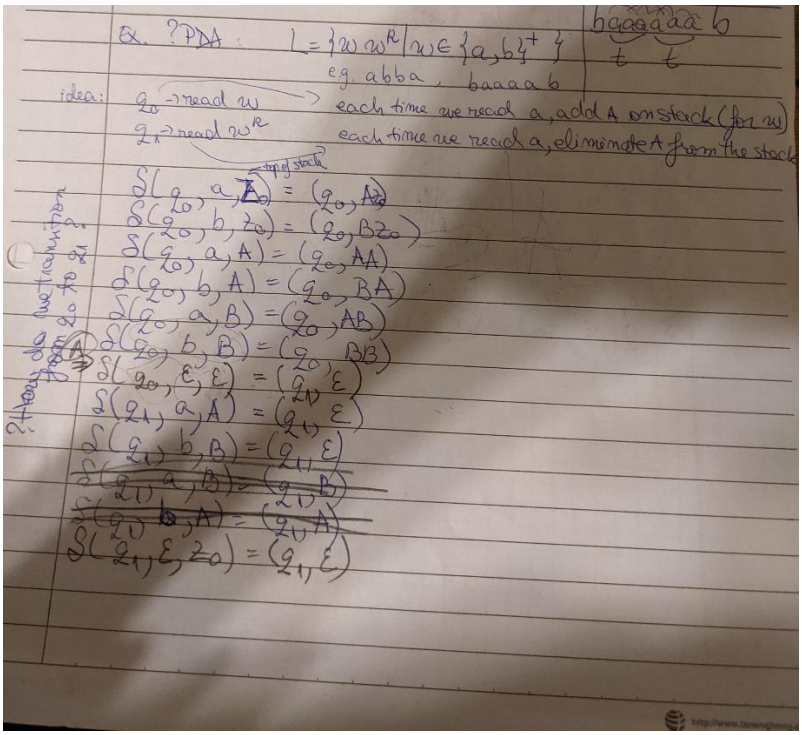
$\delta(q_1, b, B) = \{(q_1, \epsilon)\}$

$\delta(q_1, \epsilon, Z_0) = \{(q_f, Z_0)\}$ (using the final state crit.)

w=bbaabb

$(q_0, bbaabb, Z_0)$ |- $(q_0, baabb, BZ_0)$ |- |- $(q_0, aabb, BBZ_0)$ |-$(q_0, abb, ABBZ_0)$ |-(q_1, bb, BBZ_0)

|-(q_1, b, BZ_0) |-(q_1, ϵ, Z_0) |-(q_f, ϵ, Z_0) => $w \in L(M)$



2. Find a PDA that accepts the language $L = \{a^n b^{2n} | \, n \in \mathbb{N}^*\}$

$M = (\{q_0, q_1, q_f\}, \{a, b\}, \{Z_0, A\}, \delta, q_0, Z_0, \{q_f\})$

$\delta(q_0, a, Z_0) = \{(q_0, AAZ_0)\}$

$\delta(q_0, a, A) = \{(q_0, AAAZ_0)\}$

$\delta(q_0, b, A) = \{(q_1, \epsilon)\}$

$\delta(q_1, b, A) = \{(q_1, \epsilon)\}$

$\delta(q_1, \epsilon, Z_0) = \{(q_f, Z_0)\}$

HW: w = aabbbb

3. Find a PDA that accepts the language $L = \{a^{2n} b^n | \, n \in \mathbb{N}^*\}$

$M = (\{q_0, q_1, q_2, q_f\}, \{a, b\}, \{Z_0, A\}, \delta, q_0, Z_0, \{q_f\})$

$\delta(q_0, a, Z_0) = \{(q_1, Z_0)\}$

$\delta(q_1, a, Z_0) = \{(q_0, AZ_0)\}$

$\delta(q_0, a, A) = \{(q_1, A)\}$

$\delta(q_1, a, A) = \{(q_0, AA)\}$

$\delta(q_0, b, A) = \{(q_2, \epsilon)\}$

$\delta(q_2, b, A) = \{(q_2, \epsilon)\}$

$\delta(q_2, \epsilon, Z_0) = \{(q_f, Z_0)\}$

$(q_0, aaaabb, Z_0) \vdash (q_1, aaabb, Z_0) \vdash (q_0, aabb, AZ_0) \vdash (q_1, abb, AZ_0) \vdash (q_0, bb, AAZ_0) \vdash (q_2, b, AZ_0)$
$\vdash (q_2, \epsilon, Z_0) \vdash (q_f, \epsilon, Z_0)$

II. Attribute Grammars

1. Give an attribute grammar for evaluating arithmetic expressions containing *+, -, *, /, (,), id*. Validate your approach on 2*(3+4)-5.

(G,A,R)

E - > E + T $\{E_1.val = E_2.val + T.val\}$

E -> E – T $\{E_1.val = E_2.va - T.val\}$

E -> T $\{E.val = T.val\}$

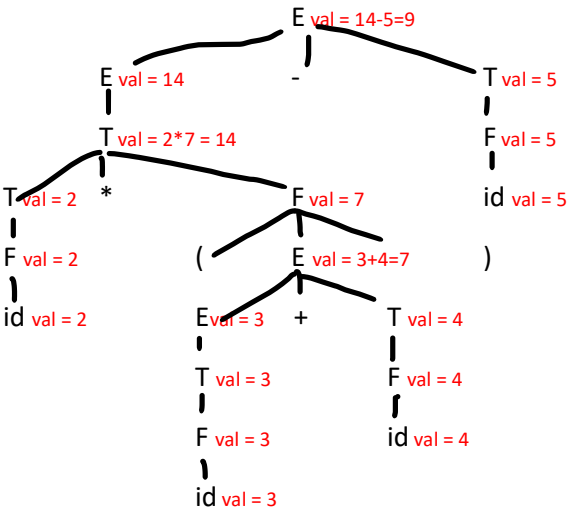
T - > T * F $\{T_1.val = T_2.val * F.val\}$

T - > T / F $\{T_1.val = T_2.val \ / \ F.val\}$

T - > F $\{T.val = F.val\}$

F – > (E) $\{F.val = E.val\}$

F -> id $\{F.val = id.val\}$



2. HW: Att. gram. for the ppf associated to an arithmetic expression

III. 3-address code

1. Represent using 3 – address code (quadruples) the following code snippet:

```
a := 4
While (a+b > 10) do
    c := c-1
d : = -1
```

index	op	Arg1	Arg2	result
1	:=	4		a
2	+	a	b	t1
3	>	t1	10	t2

4	goto	t2		(8)
5	~	1		t4
6	:=	t4		d
7	goto			(11)
8	-	c	1	t3
9	:=	t3		c
10	goto			(2)
11				