חלק תיאורטי PA2

1.נשים לב כי השפה המתקבלת על ידי האוטמט היא *(aa) + *(aaa)
דקדוק חסר הקשר לינארי ימני שמגדיר את השפה הזו הוא(S is the start symbol):

S-> D | T | ε
D -> aaD | ε
T -> aaaT | ε

2. יהי N, NFA כלשהוא. בשלב ראשון נמיר את ה-NFA ל-DFA (לצורך פשטות ההוכחה)

A = $(Q, \sum \quad , \delta, q0, F)$

כאשר q0 הוא מצב התחלתי, Q היא קבוצת המצבים, F קבוצת מצבים מקבלים, $\delta: (Q, \sum \quad) \to Q$ פונקצית מעברים.

נגדיר CFG $G = (V, \sum \quad , R, S)$ באופן הבא:

- קבוצה V תכיל משתנה B_i עבור כל מצב q_i ב-Q. כמו כן, $S = B_{q0}$ (כלומר המשתנה ההתחלתי מוגדר על ידי המצב ההתחלתי)
- הא"ב של הדקדוק הוא בדיוק הא"ב של הDFA.
- לבסוף, קבוצת הכללים R, תוגדר באופן הבא:
  לכל $q_i \in Q, a \in \sum$ נוסיף את הכלל $B_{qi} \to aB_{\delta(qi,a)}$
  כמו כן, לכל מצב מקבל $q_{accept} \in F$ נוסיף את הכלל $B_{q_{accept}} \to \varepsilon$

תחילה, קל לראות כי G הוא אכן right linear CFG. כעת נוכיח כי השפה שמוגדרת על ידי הדקדוק G היא אכן השפה שמתקבלת על ידי האוטומט A, ולכן גם על ידי האוטומט המקורי N. דהיינו נראה כי $L(G) = L(N) = L(A)$.

הוכחה:

<u>*בכיוון ראשון*</u>, יהי $w = w_1 w_2 \dots w_n \in L(A) = L(N)$, (נניח תחילה ש $w \neq \varepsilon$), אזי קיימת סדרה של מצבים $r_0, \dots, r_n \in$ Q כך שמתקיים

כי q0 = r0 וגם כי לכל $0 \leq i < n$, $\delta(r_i, w_{i+1}) = r_{i+1}$ וגם כי $r_n \in F$ (דהיינו מצב מקבל).

כעת, נראה כי w ניתנת לגזירה מהדקדוק G. לכל $0 \leq i < n$ נפעיל את הכלל הגזירה $B_{r_i} \to w_{i+1} B_{r_{i+1}}$, שאכן קיים ב-R, משום לפי הבנייה $B_{r_i} \to w_{i+1} B_{\delta(r_i, w_{i+1})} \in R$ ולפי ההנחה מתקיים $\delta(r_i, w_{i+1}) = r_{i+1}$. בפרט, עבור $i = n - 1$ נפעיל את כלל הגזירה $B_{r_{n-1}} \to w_n B_{r_n}$, ונקבל כי גזרנו את $w_1 w_2 \dots w_n B_{r_n}$ מתוך

$$B_{r0} = B_{q0} = S$$

כעת נפעיל את הכלל $B_{r_n} \to \varepsilon$, שקיים ב-R לפי הבנייה של G, משום ש-$r_n \in F$. קיבלנו גזירה של

$w_1 w_2 \dots w_n = w$ מתוך משתנה ההתחלה S, ולכן $w \in L(G)$ כרצוי.

כעת נניח כי $w = \varepsilon$, וכי $w \in L(A)$, אזי q0 הוא מצב מקבל ולכן קיים הכלל $B_{q0} = S \to \varepsilon$ ולכן $w \in L(G)$ גם במקרה זה.

<u>*בכיוון השני*</u>, נניח כי $w = w_1 w_2 .. w_n \in L(G)$, ונניח תחילה כי $w \neq \varepsilon$. נראה כי האוטומט A מקבל את w. מההנחה קיימת סדרת הפעלת כללי גזירה,

$$B_{q0} \to w_1 B_{q_{i_1}} \to w_1 w_2 B_{q_{i_2}} \to \cdots \to w_1 w_2 \dots w_n B_{q_{i_n}} \to w_1 w_2 \dots w_n$$

כאשר $q_{i_1}, q_{i_2}, \dots, q_{i_n}$ הם סדרת המצבים ב-Q המתאימים למשתנים הנ"ל. לפי הבניה של הדקדוק, מאחר ולכל $1 \leq k \leq n$ השתמשנו בכלל הגזירה $B_{q_{i_{k-1}}} \to w_k B_{q_{i_k}}$ (נסמן $q_{i_0} = q_0$), אזי בהכרח $\delta(q_{i_{k-1}}, w_k) = q_{i_k}$ לכל $1 \leq k \leq n$.

כמו כן, השתמשנו בכלל $B_{q_{i_n}} \to \varepsilon$, ולכן $q_{i_n} \in F$ לפי בניית G. לכן, לפי ההגדרה, קיבלנו שהאוטומט A מקבל את $w_1 w_2 \dots w_n = w$ ולכן $w \in L(A) = L(N)$ כנדרש.

כעת נניח כי $w = \varepsilon \in L(G)$, לכן בהכרח קיים כלל הגזירה $B_{q0} \to \varepsilon$ (כל כלל גזירה אחר מוסיף תו למילה הנגזרת, ולכן בהכרח לא ניתן לגזור את המילה הריקה ללא שימוש ישיר בכלל הנ"ל) אך לפי הבנייה משמעות הדבר כי q0 הוא מצב מקבל, ולכן $\varepsilon \in L(A)$ כרצוי.

סך הכול, הוכחנו $L(G) = L(A) = L(N)$ כאשר G הוא CFG, right linear כרצוי.

3. אכן, הדקדוק המתקבל הוא unambiguous בזכות שיטת ה- layering אך הוא מחייב את פעולת החילוק להיות אסוציאטיבית ימינה, בניגוד למקובל (חילוק היא פעולה אסוציאטיבית שמאלה).
למעשה גם הפעולות– חיבור, חיסור וכפל יהיו בעלי אסוציאטיביות ימנית בדקדוק זה,אך אלו הן פעולות אסוציאטיביות ולכן לצורך שערוך ביטויים המורכבים רק מפעולות אלה, נוכל להשתמש בעצי הגזירה המתקבלים בדקדוק זה.

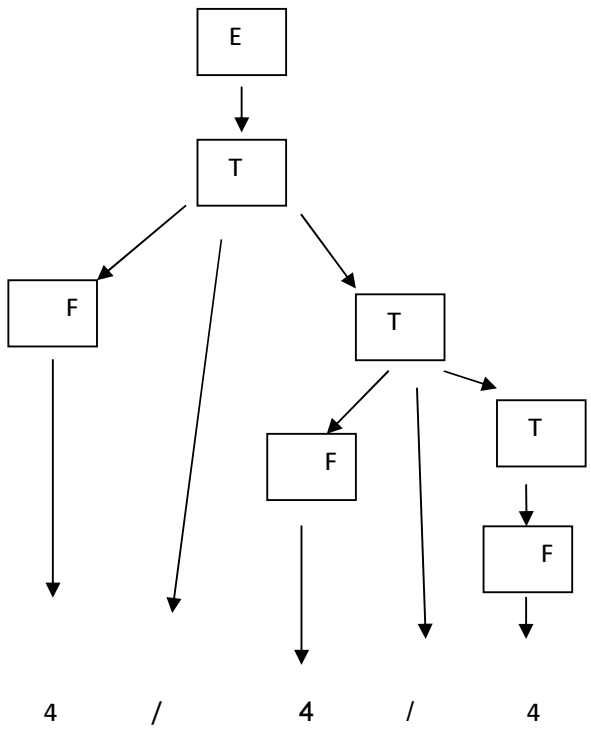
כאמור, הדקדוק מחייב את פעולות החילוק להיות אסוציאטיבית ימינה (אלא אם כן השתמשנו בסוגריים במפורש). הסיבה לכך היא שכדי לגזור חילוק אנו חייבים להשתמש בכלל

$$T \to F/T$$

וכדי להוסיף סימני חילוק נוספים לביטוי (ללא סוגריים), יש להמשיך לגזור את T באמצעות אותו הכלל (לא ניתן לגזור סימן חילוק, ללא שימוש בסוגריים, מ-F)

לדוגמא, ניקח את הביטוי 4/4/4 . השערוך המקובל של הביטוי הוא: 4/(4/4) (רבע בחילוק ממשי, אפס בחילוק שלמים). במקרה שלנו, אם נשערך את הביטוי בעזרת עץ הגזירה שמתקבל עבור הביטוי הנ"ל, בדקדוק זה, נקבל 4.

ה-parse tree המתקבל:



ניתן לראות כי הביטוי המתקבל, מבחינת סדר קדימויות, הוא 4/(4/4) בניגוד לרצוי.

4.הדקדוק אינו LL(1). נניח בשלילה כי הדקדוק הנתון הוא דקדוק LL(1). נשים לב כי שני הכללים הבאים נמצאים בדקדוק:

וגם

אזי לפי ההגדרה בהכרח מתקיים כי $First(C\ D) \cap First(c) = \emptyset$. אך מתקיים כי

$$First(c) = \{c\} \subseteq First(C) - \{\varepsilon\} \subseteq First(C\ D)$$

ולכן החיתוך הנ"ל מכיל בפרט את c, בסתירה לכך שהוא ריק.

דקדוק מתוקן:

First(S)={a,b} , first(A)={a,b} ,first(C)={c} , first(D)={d, ε},

Follow(S)={$} , Follow(A)={c} , Follow(C)={$},Follow(D)={$}.

דוגמת הרצה:

מיספור:

1)$S \to AC$
2)$A \to aA$
3)$\mathrm{A} \to b$
4)$C \to cD$
5)$D \to dD$
6)$\mathrm{D} \to \varepsilon$

: Prediction table (LL(1) parse table)

	a	b	c	d	$
S	1	1			
A	2	3			
C			4		
D				5	6

ריצה על הקלט abcdd

Input suffix	Stack content	Move
abcdd$	S$	$S \to AC$
abcdd$	AC$	$A \to aA$
abcdd$	aAC$	Match(a,a)
bcdd$	AC$	$\mathrm{A} \to b$
bcdd$	bC$	Match(b,b)
cdd$	C$	$C \to cD$
cdd$	cD$	Match(c,c)
dd$	D$	$D \to dD$
dd$	dD$	Match(d,d)
d$	D$	$D \to dD$
d$	dD$	Match(d,d)
$	D$	$\mathrm{D} \to \varepsilon$
$	$	Match($,$)- **success**

5.(i) נאסוף את כל אברי LR0 מתוך מצבי האוטומט המתואר, נוריד את ההקשר (סימן הנקודה) ונקבל את אוסף הכללים של הדקדוק.

נקבל את ה-CFG הבא, כאשר S' הוא ה-start symbol:

S' -> S

S -> bAb

A -> a | (B

B -> Aa)

(ii) תפקיד ה-CFSM <u>אינו</u> לקבל את השפה שמוגדרת על ידי הדקדוק (ניתן גם לראות שהוא מכיל "סימנים" (non terminals) שהם אינם חלק מהא"ב של השפה) ולכן ייתכן שהמילה היא בשפה שמוגדרת על ידי הדקדוק, אך אין לה ריצה מקבלת ב-CFSM. כך למשל במקרה שלנו, אין ריצה

מקבלת (כלומר מסתיימת במצב מקבל, במקרה שלנו, מצב reduce) של ה-CFSM על המילה "bab".
כמו כן נעיר כי אילו $S \to bab$ היה כלל בדקדוק שלנו, הרי שכן היה ב- CFSM המתאים לדקדוק החדש מסלול מקבל על המילה "bab" (שכן היינו מתחילים מהמצב ההתחלתי שהיה מכיל את האיבר $S \to .bab$, רואים b, מגיעים למצב שמכיל את האיבר $S \to b.ab$, כעת רואים a, מגיעים למצב שמכיל את האיבר $S \to ba.b$ ולבסוף רואים את b ומגיעים למצב שמכיל $S \to bab.$ שהוא מצב reduce ולכן במקרה שלנו "מקבל")

ה-CFSM מתאר את טבלת המעברים (action and goto) של ה-PDA (של ה- LR0 parser), דהיינו, הוא מתאר עבור כל מצב (שמתאר את ההקשר שבו אנחנו גוזרים כעת את המשפט, דהיינו, מה ראינו עד כה ומה אנחנו מצפים לראות ולפי אילו כללי גזירה) ועבור כל token או non-terminal מהו המצב הבא (ב-CFSM) שיש לדחוף למחסנית (shift) או לחילופין האם צריך לעשות pop מראש המחסנית ולעשות reduce ל- non terminal לפי הכלל שעשינו לו pop.

המילה bab מתקבלת על ידי ה-PDA (*ולא ה-CFSM*) באופן הבא:

- הפרסר ידחוף את הזוג (b, I1) למחסנית. [קלט כעת: ab]
- הפרסר ידחוף את הזוג (a, I5) למחסנית. [קלט כעת: b]
- מצב reduce נמצא בראש המחסנית (I5), עושים pop ל-(a, I5), דוחפים A למחסנית (reduce). [קלט: b]
- המצב בראש המחסנית לפני שדוחפים את A הוא I1, לכן, מבצעים GOTO (דוחפים את המצב I2 למחסנית) [קלט: b]
- דוחפים את (b, I3) לראש המחסנית (קלט כעת ריק)
- מצב reduce בראש המחסנית, מוציאים את (b,I3),(A, I2)(b, I1) מהמחסנית, בראש המחסנית כעת המצב I0 + דוחפים את S.
- מבצעים GOTO (מצב I0, קיבלנו nonterminal S), דהיינו דוחפים את I4 למחסנית
- I4 הוא מצב מקבל [שכן הקלט ריק וקיבלנו את ה- start symbol S'], לכן bab נמצאת בשפה שמוגדרת על ידי הCFG.

(iii) בתיאור ה-Stack גודל לכיוון ימין, ראש הקלט נמצא בצד שמאל.

Stack	**Input**	**Action**
I0	b ((a a) a) b$	shift I1
I0, (b, I1)	((a a) a) b$	shift I6
I0, ('b', I1), ('(', I6)	(a a) a) b	shift I6
I0, ('b',I1),('(', I6),('(',I6)	a a) a) b$	shift I5
I0, ('b',I1),('(', I6),('(',I6),(a,I5)	a) a) b$	Reduce A->a
I0, ('b',I1),('(',I6),('(,I6),(A,I8)	a) a) b$	shift I9
I0, ('b',I1),('(', I6),('(',I6),(A,I8),(a,I9)	) a) b$	shift I10

I0, (‘b’,I1),(‘(‘, I6),(‘(‘,I6),(A,I8),(a,I9),(‘)’,I10)	a) b $	Reduce B ->Aa)
I0, (‘b’,I1),(‘(‘, I6),(‘(‘,I6),(B,I7)	a) b$	Reduce A ->(B
I0, (‘b’,I1),(‘(‘, I6),(A,I8)	a)b$	shift I9
I0, (‘b’,I1),(‘(‘, I6),(A, I8),(a, I9)	)b$	shift I10
I0, (‘b’,I1),(‘(‘, I6),(A,I8),(a,I9),(‘)’, I10)	b$	Reduce B->Aa)
I0, (‘b’, I1), (‘(‘, I6), (B, I7)	b$	Reduce A->(B
I0, (‘b’, I1), (A, I2)	b$	shift I3
I0, (‘b’, I1), (A, I2), (‘b’, I3)	$	Reduce S->bAb
I0, (S, I4)	$	**accept**