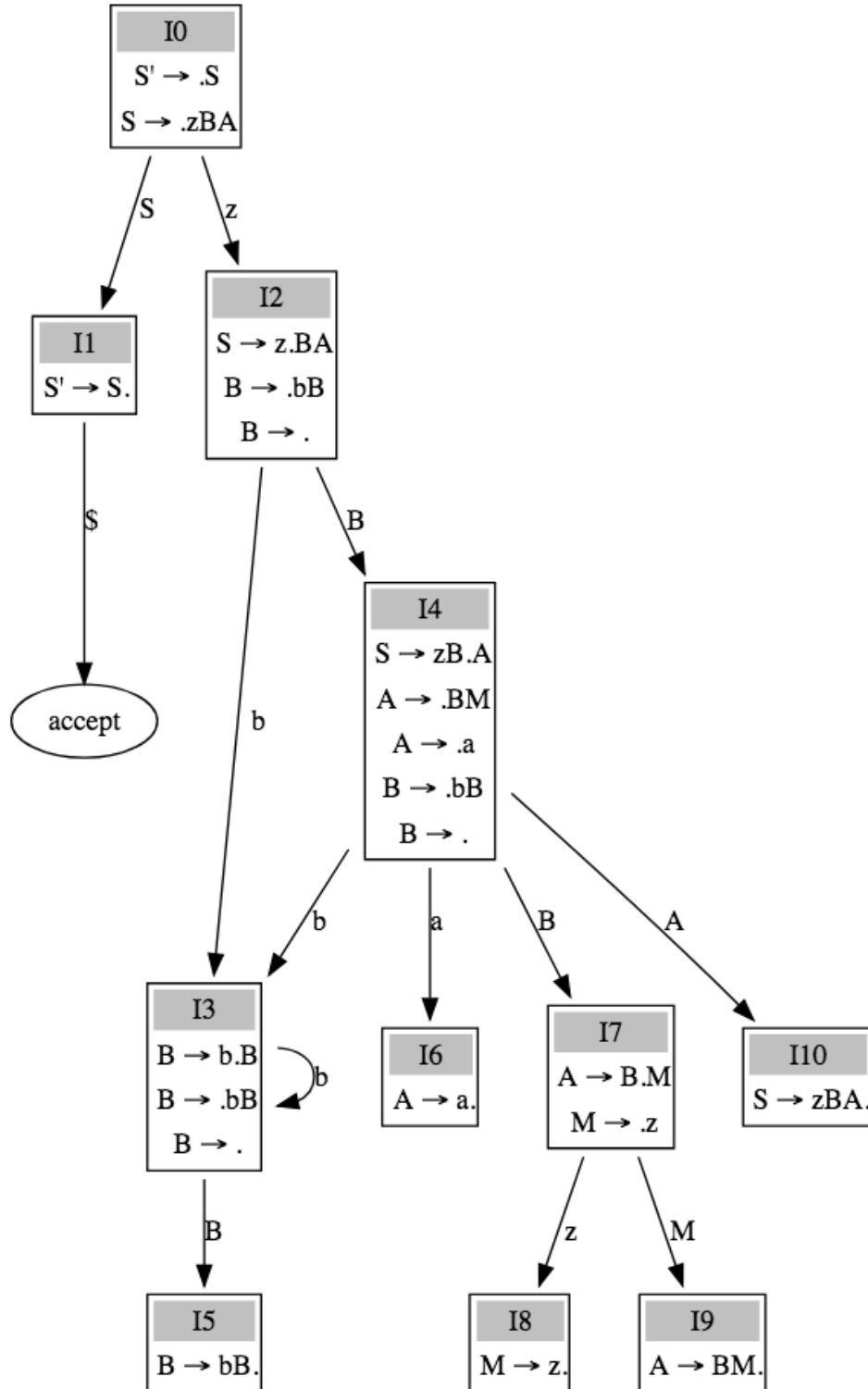


שאלה 1

סעיף א':

עבור הדקדוק הנתון, נבנה אוטומט פריטי LR0 - הרחבתי את הדקדוק על ידי הוספת כלל התחלתי $S' \rightarrow S$ וכעת המשתנה ההתחלתי הינו S :

$S' \rightarrow S, S \rightarrow zBA, B \rightarrow bB \mid \varepsilon, A \rightarrow BM \mid a, M \rightarrow z$



סעיף ב':

כעת אבנה טבלת פיסוק SLR1 עבור הדקדוק מסעיף א' לפי האלגוריתם¹ לבניית טבלת פיסוק SLR1:
לצורך הנוחות נכתוב את כללי הגזירה כרשימה:

1. $S' \rightarrow S$
2. $S \rightarrow zBA$
3. $A \rightarrow BM$
4. $A \rightarrow a$
5. $B \rightarrow bB$
6. $B \rightarrow \varepsilon$
7. $M \rightarrow z$

	a	b	z	\$	S'	S	A	B	M
0			s2			1			
1				accept					
2	r6	s3, r6	r6					4	
3	r6	s3, r6	r6					5	
4	s6, r6	s3, r6	r6				10	7	
5	r5	r5	r5						
6				r4					
7			s8						9
8				r7					
9	r3	r3	r3						
10				r2					

נחשב את Follow של משתני הדקדוק:

$$Follow(A) = \{\$ \}$$

$$Follow(B) = \{a, b, z\}$$

$$Follow(M) = \{\$ \}$$

$$Follow(S) = \{\$ \}$$

סעיף ג': בניית טבלת מעקב עבור ה-parser² של הדקדוק הנתון, עבור הקלט: acb.

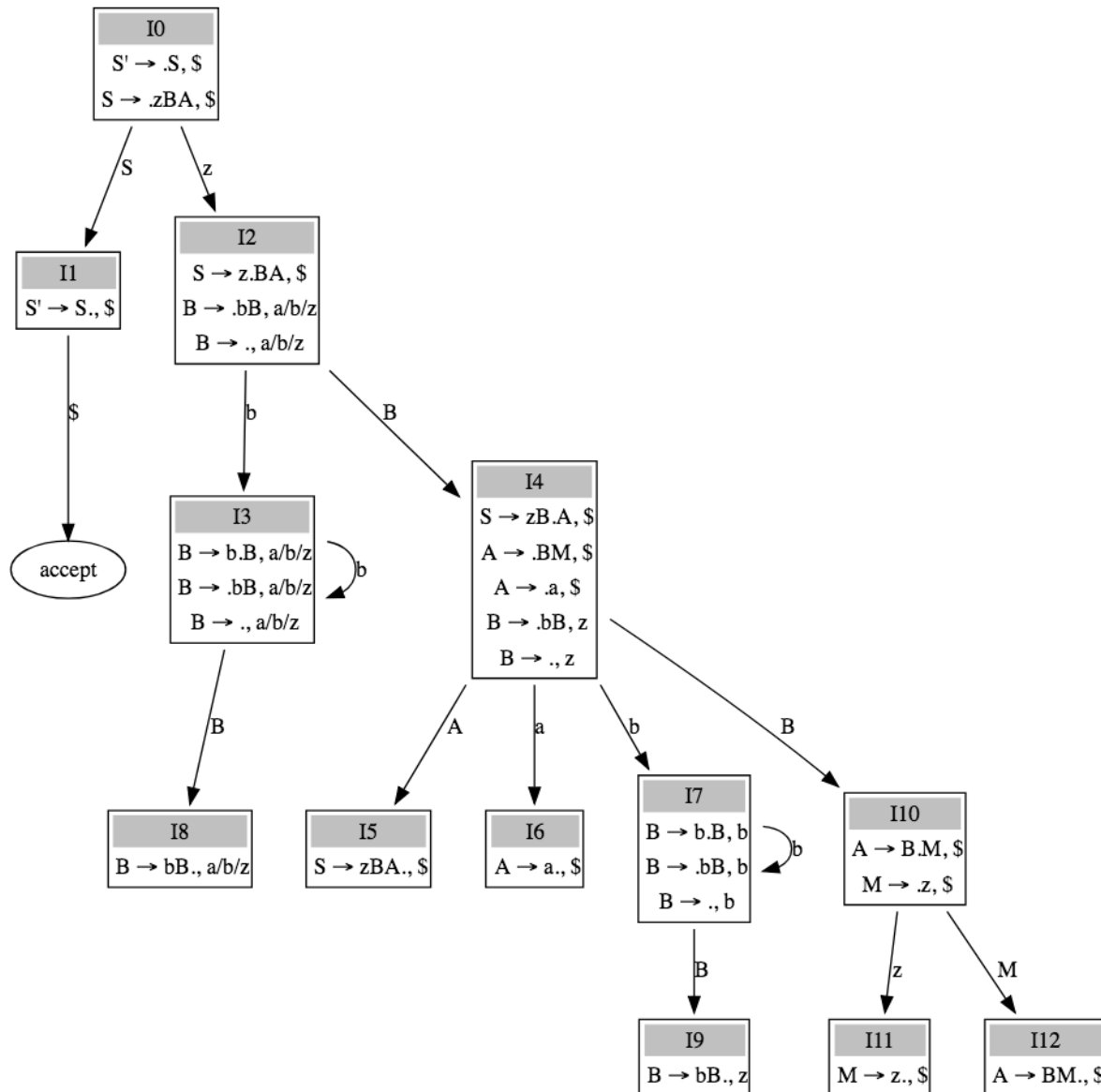
Stack	Input	Action
-------	-------	--------

¹ אלגוריתם 4.46, עמוד 253 בספר הלימוד.

² אלגוריתם 4.44, עמוד 251 בספר הלימוד.

0	acb\$	shift 3
0 a 3	cb\$	reduce $C \rightarrow \varepsilon$ (note $ \varepsilon = 0$)
0 a 3 C 7	cb\$	shift 8
0 a 3 C 7 c 8	b\$	reduce $C \rightarrow Cc$ ($goto(3, C)$)
0 a 3 C 7	b\$	reduce $A \rightarrow aC$
0 A 2	b\$	shift 6
0 A 2 b 6	\$	reduce $B \rightarrow b$
0 A 2 B 5	\$	reduce $S \rightarrow AB$
0 S 1	\$	accept

סעיף ד': יש לבנות אוטומט פריטי LR1 (עבור הדקדוק מסעיף א').



נשים לב כי בדקדוק זה:

1. אין reduce-reduce conflict
 2. נשים לב כי קיים shift reduce conflict במצב I2, שכן $[B \rightarrow .bB, a/b/z]$ וגם $[B \rightarrow ., b]$ נמצאים במצב I2.
- לכן דקדוק זה איננו ב-LR1.

סעיף ה':

נחשב את כמות המצבים של דקדוק זה עבור מפרש LALR1: נתבונן במכונת המצבים מסעיף קודם, המצבים אשר ניתן לאחד הם:

1. מצבים I3, I7
2. מצבים I8, I9

לכן יהיו 11 מצבים למכונת המצבים.

שאלה 3

נתון הדקדוק G בעל כללי הגזירה הבאים:

$$S \rightarrow cABc \mid Ba$$

$$A \rightarrow acA \mid bbSA \mid cbS$$

$$B \rightarrow bA \mid \varepsilon$$

אכתוב recursive decent parser עבור דקדוק זה (אתבסס על הפסאודו-קוד המתואר בספר³ עם שינויים קלים - במקום לבצע backtracking, אני אבדוק את ה-prefix הנוכחי כנגד כללי הדקדוק ואם זה אכן מתאים לכלל - אמשיך לפרש את הביטוי באופן רקורסיבי, אחרת התוכנית תצא עם הודעת שגיאה. למעשה, אפשר לומר כי המימוש שלי הינו recursive descent parser עם lookahead של 2, זה לצורך הנוחות - כדי להימנע מ-backtracking מורכב).

ראשית, נעזר בטבלת המעברים מהממ"ן הקודם:

	a	b	c	\$
S	$S \rightarrow Ba$	$S \rightarrow Ba$	$S \rightarrow cABc$	
A	$A \rightarrow acA$	$A \rightarrow bbSA$	$A \rightarrow cbS$	
B	$B \rightarrow \varepsilon$	$B \rightarrow bA$	$B \rightarrow \varepsilon$	

מצורף קוד חלקי בשפת פייתון המממש Recursive descent parser עבור הדקדוק הנתון, עם הערות ליד כל תנאי איזה ביטוי הוא מבטא - **הוספתי את קוד הפייתון המקורי בקובץ נפרד**. נניח כי INPUT הוא משתנה גלובאלי המייצג את הקלט הנתון. נניח כי error היא פונקציה היוצאת מהתכנית במקרה של שגיאה.

נתחיל מהפונקציה A:

```
def A():
    if lookahead() == 'a': #  $A \rightarrow acA$ 
        match("a")
        match("c")
        A()
    elif lookahead() == 'b': #  $A \rightarrow bbSA$ 
        match('b')
        match('b')
        S()
        A()
    elif lookahead() == 'c': #  $A \rightarrow cbS$ 
        match('c')
        match('b')
        S()
    else:
```

³ פרק 4.4.1 עמוד 219 בספר הלימוד.

```
error()
```

הפונקציה S:

```
def S():
    if lookahead() == "c": #  $S \rightarrow cABc$ 
        match('c')
        A()
        B()
        match("c")
    elif lookahead() in ('a', 'b'): #  $S \rightarrow Ba$ 
        B()
        match("a")
    else:
        error()
```

הפונקציה B:

```
def B():
    if lookahead() == "b": #  $B \rightarrow bA$ 
        match("b")
        A()
    # This is the rule  $B \rightarrow \epsilon$ .
    elif lookahead() in ('a', 'c'):
        pass
    else:
        error()
```

שאלה 4

נתון הדקדוק הבא (אמספר את החוקים לצורך הנוחות):

1. $S \rightarrow 1A0$
2. $S \rightarrow B1$
3. $S \rightarrow 2$
4. $A \rightarrow 0A$
5. $A \rightarrow 2B0$
6. $B \rightarrow 2B$
7. $B \rightarrow 1$
8. $B \rightarrow \epsilon$

סעיף א': יש להראות כי הדקדוק הנתון איננו שייך ל-LL1:

נתבונן על כללי הגזירה של S ובפרט: $S \rightarrow 1A0 \mid B1$. נחשב את ה-First של שני כללים אלו:

$First(1A0) = \{1\}$, $First(B1) = \{1, 2, \epsilon\}$ מכאן קל לראות כי $First(1A0) \cap First(B1) = \{1\}$ ולכן

דקדוק זה איננו שייך ל-LL1 (לו היינו בונים טבלת פיסוק, היה תא המכיל 2 כללי גזירה וזה מצב לא חוקי).

סעיף ב': האם דקדוק זה שייך ל-LL2?

נבנה טבלת פיסוק עבור 2 טרמינלים, כאשר ערך בתא יהיה מספר כלל הגזירה - לפי המיספור המופיע למעלה.

	0\$	00	01	02	1\$	10	11	12	2\$	20	21	22	\$
S					2	1	2	1	3		2	2	
A		4		4						5	5	5	
B		8			7,8				6		6	6	

נשים לב כי עבור הכלל $\varepsilon \rightarrow B$ (כלל מספר 8) אהיה צריך חשב את $Follow_2(B)$:

$$Follow_2(B) = \{1$, 00\}$$

נשים לב כי בתא $M[B, 1$]$ יש שני כללים, ולכן הדקדוק הנתון איננו ב-LL2.