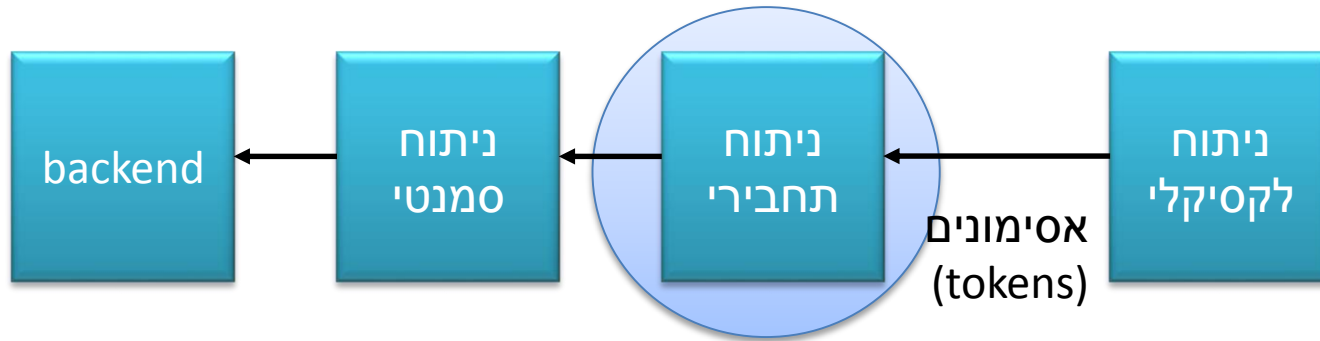


תזכורת מתרגולים אחרונים

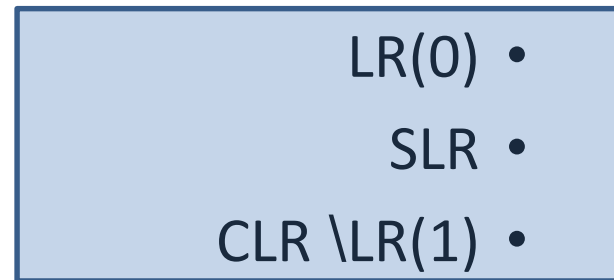
- מבנה סכמתי של קומפיילר



- ניתוח תחבירי:

– Top Down : RD , LL(1)

– Bottom up :



• LALR

מנתח LR

ניתוח מלמטה למעלה – Bottom Up

- מתחילים את הניתוח במילת הקלט, ומנסים להגיע עד למשתנה ההתחלתי (שורש עץ הגזירה).
- מחליפים תבנית פסוקית במשתנה הגוזר אותה.

משמעות LR:

- קוראים את הקלט **משמאל לימין (L)**
- מוצאים את הגזירה **הימנית (R)** ביותר, בסדר **הפוך** לסדר הגזירה.

- (1) $S \rightarrow A B$
- (2) $A \rightarrow \underline{a} \underline{a} \underline{a} C$
- (3) $B \rightarrow \underline{b} \underline{b} \underline{b}$
- (4) $C \rightarrow \underline{c} \underline{c}$

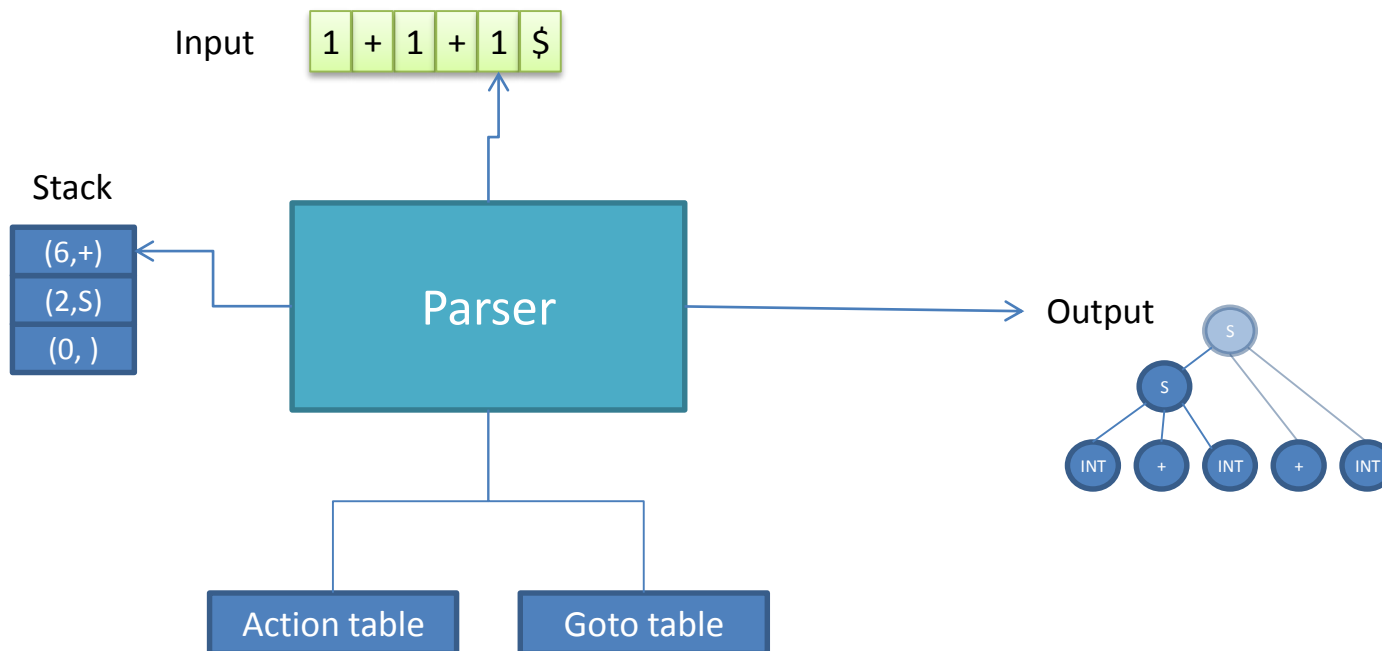
דוגמא לניתוח בגישת Bottom-Up:
מילת הקלט: $w = \underline{a} \underline{a} \underline{a} \underline{c} \underline{c} \underline{b} \underline{b} \underline{b}$

$\underline{a} \underline{a} \underline{a} \underline{c} \underline{c} \underline{b} \underline{b} \underline{b}$	\Rightarrow	$\underline{a} \underline{a} \underline{a} C \underline{b} \underline{b} \underline{b}$	\Rightarrow	$A \underline{b} \underline{b} \underline{b} $	\Rightarrow	$A B $	\Rightarrow	$S $	כיוון הניתוח
\Leftarrow		\Leftarrow		\Leftarrow		\Leftarrow			כיוון הגזירה

מבנה מנתח LR

מנתח LR מכיל:

- (1) **מחסנית:** תוכנה מייצג את התבנית הפסוקית שראינו עד עתה.
הניתוח ייעצר כאשר תוכנה יהיה רק S – שורש עץ הגזירה.
- (2) **טבלת ניתוח** התלויה בדקדוק (goto - action).



בניית מנתח LR

שלבי בניית המנתח:

- (1) בניית אוטומט פרפיקסי לפי הדקדוק הנתון
- (2) בניית טבלת הניתוח ע"פ האוטומט הפרפיקסי

הרצת המנתח על קלט:

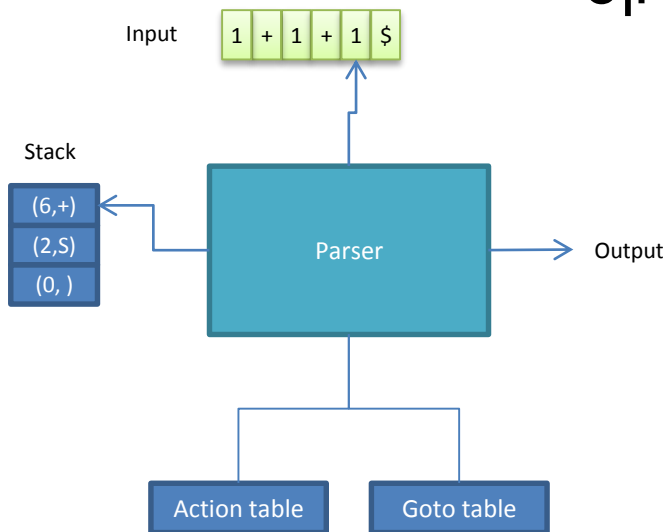
בכל צעד המנתח יכול:

- להכניס תו מהקלט למחסנית (**Shift**)

או:


- לצמצם תבנית פסוקית בראש המחסנית

למשתנה הגוזר אותה (**Reduce**)



כלומר: בכל פעם שמגיעים לתת-עץ בסריקת הקלט, בונים אותו.

מנתח LR(0) – בניית האוטומט

• אין lookahead: 
מצפים מהאלגוריתם לזהות כל כלל לאחר קריאת כל החלק הימני שלו, בלי קריאת ההמשך.

• פריט LR(0) הוא $(A \rightarrow \alpha \bullet \beta)$ כאשר $A \rightarrow \alpha \beta \in P$

- כל אחד ממצבי האוטומט המנתח הוא קבוצת פריטי LR(0).
- פריט מסמל את מצבו של המנתח.
- משמעותו: זיהינו את מה שנמצא לפני הנקודה, וכעת אנו מצפים למצוא את מה שנמצא מימינה.

שאלה: אילו פריטים אפשר לקבל עבור הכלל: $A \rightarrow \varepsilon$?

מנתח LR(0) – בניית האוטומט

- פריט LR(0) הוא $(A \rightarrow \alpha \bullet \beta)$ כאשר $A \rightarrow \alpha \beta \in P$
- כל אחד ממצבי אוטומט המנתח הוא קבוצת פריטי LR(0).

- סגור (closure) על קבוצת פריטים I מוגדר באופן אינדוקטיבי:

• בסיס: $\text{closure}(I) = I$

• צעד: אם $(A \rightarrow \alpha \bullet B \beta) \in \text{closure}(I)$ אז

לכל $B \rightarrow \gamma \in P$ גם $(B \rightarrow \bullet \gamma) \in \text{closure}(I)$

קבוצת פריטי
LR(0)

$X \in T \cup V$

• פונקצית המעברים של האוטומט ממצב ימין X :

$$\delta(I, X) = \bigcup \left\{ \text{closure}(A \rightarrow \alpha X \bullet \beta) \mid (A \rightarrow \alpha \bullet X \beta) \in I \right\}$$

מנתח LR(0) – בניית האוטומט

לפני בניית האוטומט מוסיפים לדקדוק כלל $S' \rightarrow \cdot S$

אלגוריתם בניית האוטומט

אתחול: המצב I_0 יסמן את המצב ההתחלתי של האוטומט, ויוגדר כ –

$$I_0 = \text{Closure}(\{S' \rightarrow \cdot S\})$$

כל עוד קיים מצב שלא פותח:

1. בוחרים מצב שלא פותח (I) .

2. לכל סימן $X \in V \cup T$ עבורו קיים פריט $i \in I$ מהצורה $i = A \rightarrow \alpha \cdot X\beta$

1. מחשבים את $\delta(I, X)$ (ומוסיפים אותו לקב' המצבים, אם עדיין לא חלק ממנה)

2. יוצרים קשת עם הסימן X , שמובילה למצב $\delta(I, X)$.

0) $S' \rightarrow S$

1) $S \rightarrow aA$

2) $S \rightarrow aB$

3) $A \rightarrow a$

4) $B \rightarrow b$

דוגמא - LR (0)

• בניית אוטומט פרפיקסי:

0

$S' \rightarrow \bullet S$

$S \rightarrow \bullet aA$

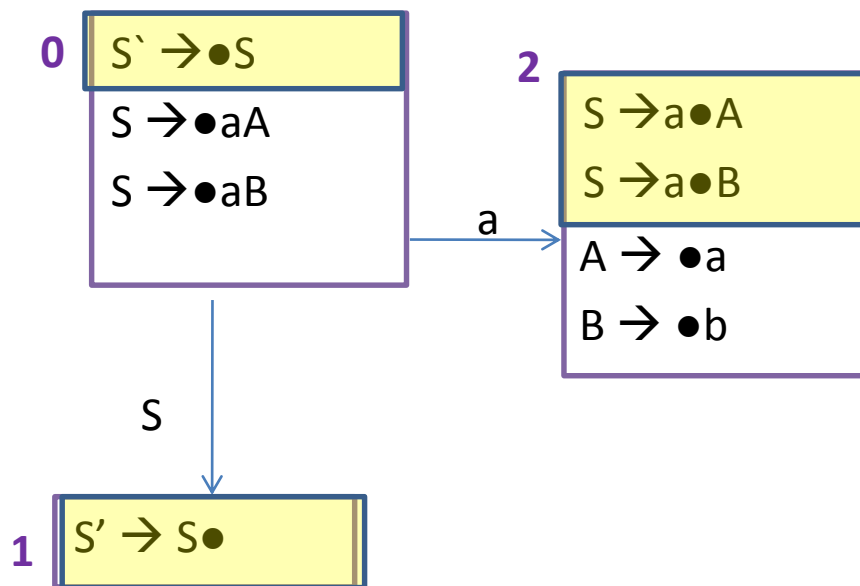
$S \rightarrow \bullet aB$

$$I_0 = \text{closure}(\{S' \rightarrow \bullet S\})$$

- 0) $S' \rightarrow S$
- 1) $S \rightarrow aA$
- 2) $S \rightarrow aB$
- 3) $A \rightarrow a$
- 4) $B \rightarrow b$

דוגמא - LR (0)

• בניית אוטומט פרפיקסי:



1. לכל סימן $X \in V \cup T$ עבורו קיים פריט $A \rightarrow \alpha \cdot X \beta \in I$:

1. מחשבים את $\delta(I, X)$ (ומוסיפים אותו לקב' המצבים, אם עדיין לא חלק ממנה)

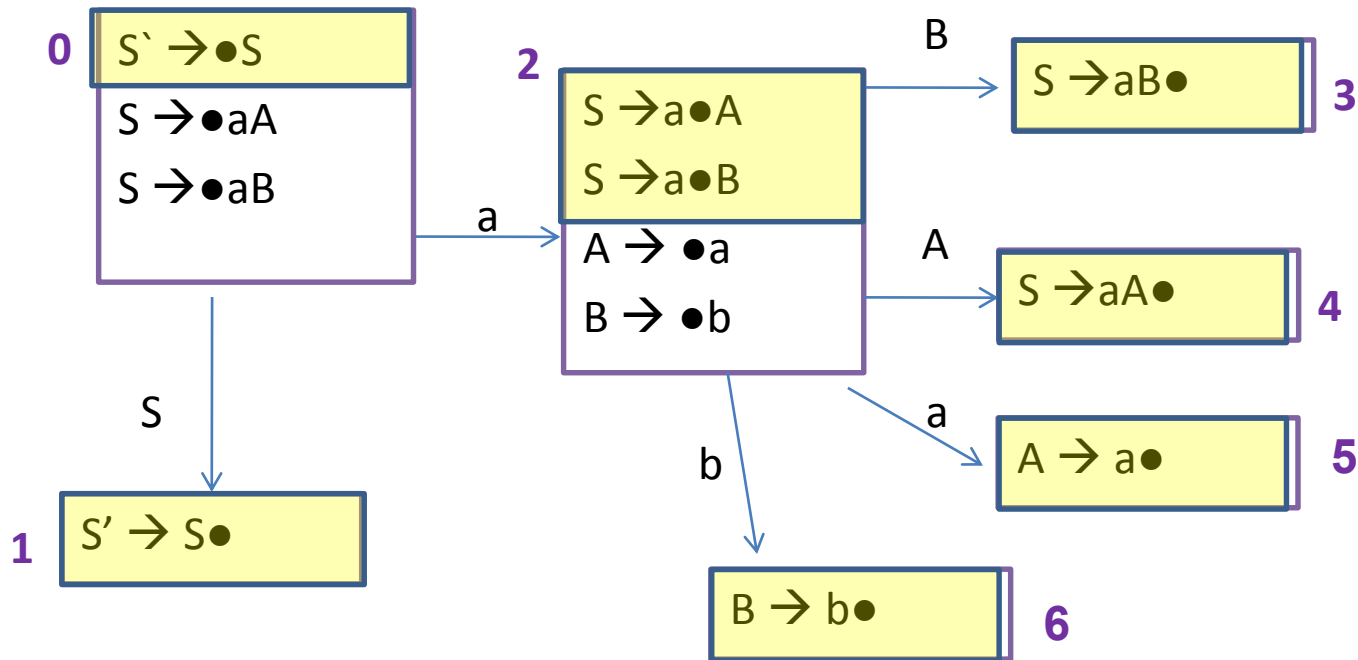
2. יוצרים קשת עם הסימן X , שמובילה למצב $\delta(I, X)$.

$$\delta(I, X) = \bigcup \left\{ \text{closure}(A \rightarrow \alpha X \bullet \beta) \mid (A \rightarrow \alpha \bullet X \beta) \in I \right\}$$

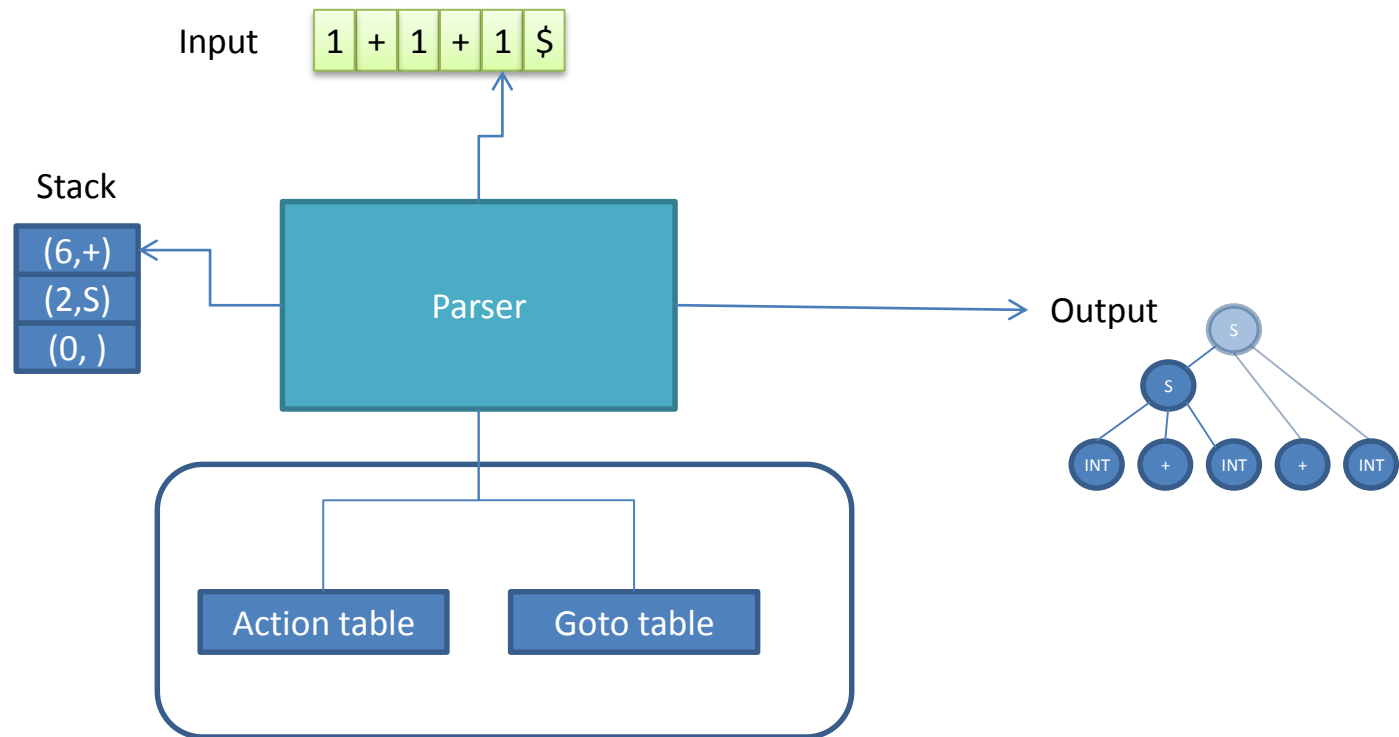
- 0) $S' \rightarrow S$
- 1) $S \rightarrow aA$
- 2) $S \rightarrow aB$
- 3) $A \rightarrow a$
- 4) $B \rightarrow b$

LR (0) - דוגמא

• בניית אוטומט פרפיקסי:



מנתח LR(0) – בניית טבלת הניתוח



מנתח LR(0) – בניית טבלת הניתוח

הגדרת טבלת action למנתח LR(0):

אינדקס של
מצב באוטומט

$t \in T$

$\text{action}[i, t] =$

$$\begin{cases} \text{SHIFT}_j & \delta(I_i, t) = I_j \\ \text{REDUCE}_k & \text{rule } k \text{ is } A \rightarrow \alpha, (A \rightarrow \alpha \bullet) \in I_i \\ \text{ACCEPT} & (S' \rightarrow S \bullet) \in I \text{ and } t = \$ \\ \text{ERROR} & \text{otherwise} \end{cases}$$

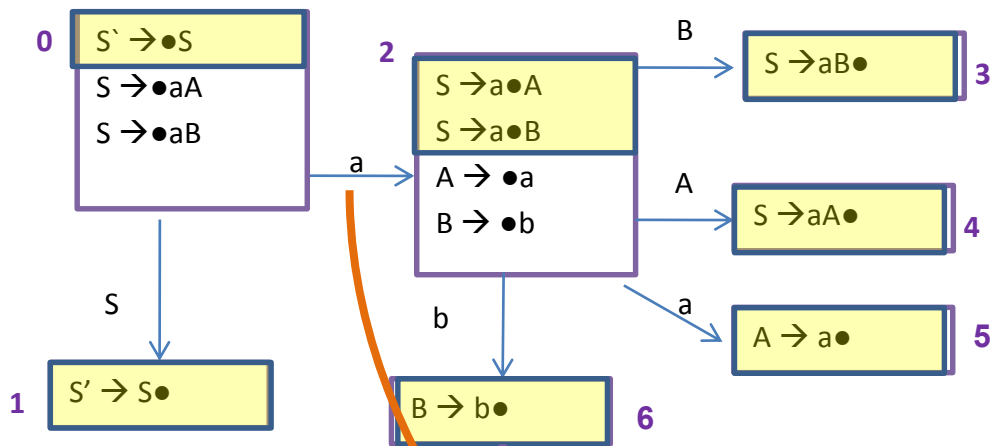
הגדרת טבלת goto למנתח LR(0):

$X \in V$

$\text{goto}[i, X] =$

$$\begin{cases} j & \delta(I_i, X) = I_j \\ \text{error} & \text{otherwise} \end{cases}$$

דוגמא – בניית טבלת הניתוח



מספר המצב אליו עוברים

- 0) $S' \rightarrow S$
- 1) $S \rightarrow aA$
- 2) $S \rightarrow aB$
- 3) $A \rightarrow a$
- 4) $B \rightarrow b$

מספר הכלל שגוזרים

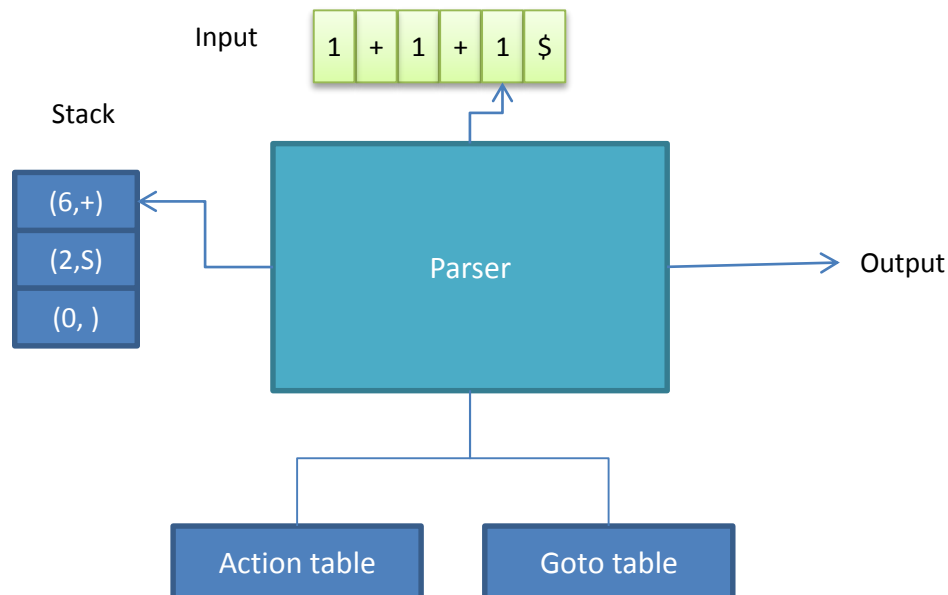
actions			
	a	b	\$
0	s2		
1			acc
2	s5	s6	
3	r2	r2	r2
4	r1	r1	r1
5	r3	r3	r3
6	r4	r4	r4

goto		
S	A	B
1		
	4	3

אלגוריתם הניתוח

Parser:

```
Q.push( (0, ) )      //where 0 is the initial state pf the prefix automaton
while true do
    k = Q.top().state
    t = next token
    do action[k, t]
end while
```



- 0) $S' \rightarrow S$
- 1) $S \rightarrow aA$
- 2) $S \rightarrow aB$
- 3) $A \rightarrow a$
- 4) $B \rightarrow b$

דוגמא LR (0) - הרצה

רצף האסימונים בקלט: $ab\$$

	a	b	\$
0	s2		
1			acc
2	s5	s6	
3	r2	r2	r2
4	r1	r1	r1
5	r3	r3	r3
6	r4	r4	r4

Action

	S	A	B
0	1		
1			
2			3
3			
4			
5			
6			

goto

קלט	מחסנית	פעולה
ab\$	(0,)	Action[0,a] = Shift 2
b\$	(0,), (2,a)	Action[2,b] = Shift 6

• **shift(k)** - (בצע shift אל מצב k):

1. דחוף למחסנית את (k, \underline{t}) .

2. קדם את הראש הקורא את הקלט צעד אחד ימינה.

- 0) $S' \rightarrow S$
- 1) $S \rightarrow aA$
- 2) $S \rightarrow aB$
- 3) $A \rightarrow a$
- 4) $B \rightarrow b$

דוגמא LR (0) - הרצה

רצף האסימונים בקלט: $ab\$$

	a	b	\$
0	s2		
1			acc
2	s5	s6	
3	r2	r2	r2
4	r1	r1	r1
5	r3	r3	r3
6	r4	r4	r4

Action

	S	A	B
0	1		
1			
2			3
3			
4			
5			
6			

goto

קלט	מחסנית	פעולה
$ab\$$	$(0,)$	Action[0,a] = Shift 2
$b\$$	$(0,), (2,a)$	Action[2,b] = Shift 6
$\$$	$(0,), (2,a), (6,b)$	Action[6,\$] = Reduce (4)
$\$$	$(0,), (2,a), (3,B)$	

reduce(j) - (בצע reduce לפי $A \rightarrow \alpha$, כלל הגזירה שמספרו הוא j):

1. הוצא $|\alpha|$ זוגות מהמחסנית. סמן ב k' את המצב שהתגלה בראש

המחסנית.

2. דחוף למחסנית את $(goto[k', A], A)$.

3. ניתן להוציא כפלט את j (מספר כלל הגזירה בו השתמשנו). הדפסת כללי

הגזירה בסדר הפוך תיתן את הגזירה הימנית ביותר.

- 0) $S' \rightarrow S$
- 1) $S \rightarrow aA$
- 2) $S \rightarrow aB$
- 3) $A \rightarrow a$
- 4) $B \rightarrow b$

דוגמא LR (0) - הרצה

רצף האסימונים בקלט: $ab\$$

	a	b	\$
0	s2		
1			acc
2	s5	s6	
3	r2	r2	r2
4	r1	r1	r1
5	r3	r3	r3
6	r4	r4	r4

Action

	S	A	B
0	1		
1			
2			3
3			
4			
5			
6			

goto

קלט	מחסנית	פעולה
$ab\$$	$(0,)$	Action[0,a] = Shift 2
$b\$$	$(0,), (2,a)$	Action[2,b] = Shift 6
$\$$	$(0,), (2,a), (6,b)$	Action[6,\$] = Reduce (4)
$\$$	$(0,), (2,a), (3,B)$	Action[3,\$] = Reduce (2)
$\$$	$(0,), (1,S)$	Action[1,\$] = accept

**מדוע מוסיפים את כלל 0 ?

<< על מנת שהמשתנה ההתחלתי יופיע בחוקי הגזירה רק באגף שמאל.
 כך בצמצום למשתנה ההתחלתי אכן יובטח שאין עוד סימנים
 במחסנית ונדע שהסתיימה הגזירה.

קונפליקטים אפשריים במנתחי LR

• קונפליקט נוצר בטבלת הניתוח כאשר יש 2 פעולות שונות או יותר באותה משבצת בטבלה.

• קיימים 2 סוגי קונפליקטים:

(1) קונפליקט shift/reduce :

$$\begin{array}{l} A \rightarrow \alpha \bullet t \beta \\ B \rightarrow \delta \bullet \end{array}$$

האם להמשיך לקרוא את הקלט או לצמצם ?

(2) קונפליקט reduce/reduce :

$$\begin{array}{l} A \rightarrow \alpha \bullet \\ B \rightarrow \delta \bullet \end{array}$$

לפי איזה כלל גזירה לצמצם ?

שאלה: מדוע אין קונפליקט shift/shift ?

מנתח SLR

- בשביל להיפטר מקונפליקטים, נרצה להכניס הסתכלות על התו הראשון של הקלט
- האם ההחלטה (shift, reduce) הגיונית בהקשר?
- ההקשר: מה יכול לקרות אחרי המשתנה הנוכחי
- כלי אפשרי מתרגול קודם: follow

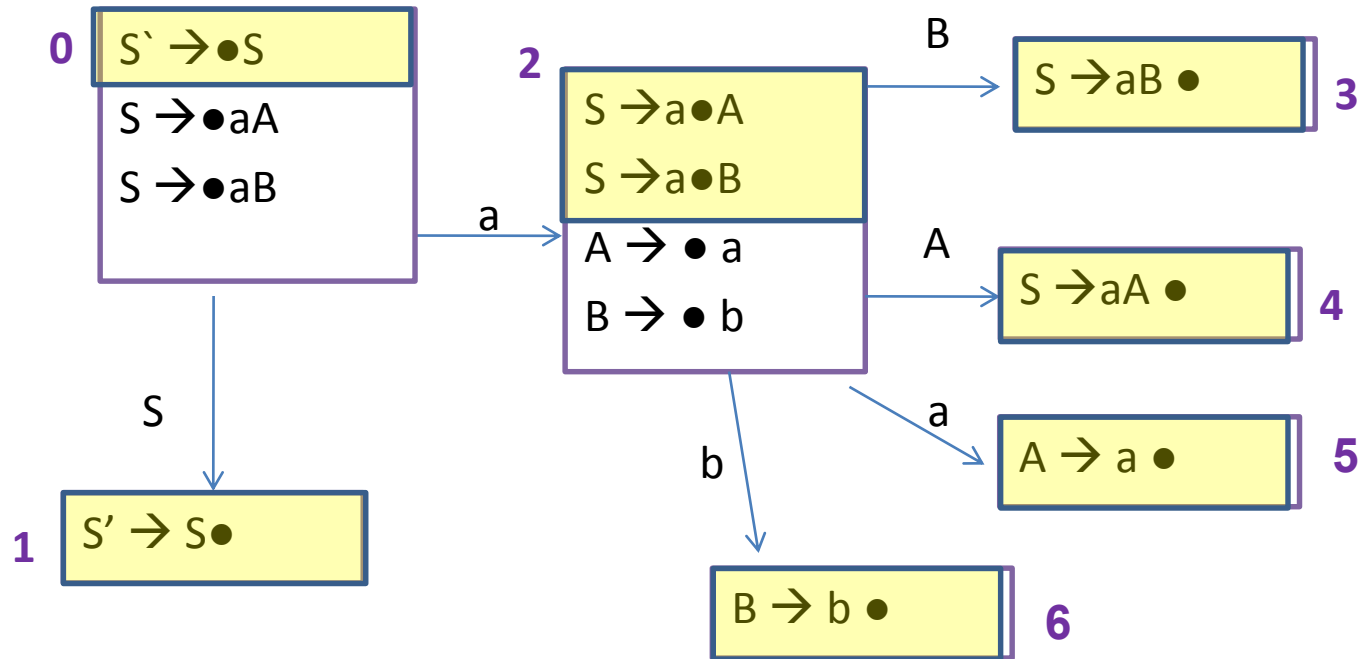
הגדרת טבלת action למנתח SLR:

$$\text{action}[i, t] = \begin{cases} \text{SHIFT}_j & \delta(I_i, t) = I_j \\ \text{REDUCE}_k & \text{rule } k \text{ is } A \rightarrow \alpha, (A \rightarrow \alpha\bullet) \in I_i \text{ and } t \in \text{follow}(A) \\ \text{ACCEPT} & (S' \rightarrow S\bullet) \in I_i \text{ and } t = \$ \\ \text{ERROR} & \text{otherwise} \end{cases}$$

- 0) $S' \rightarrow S$
- 1) $S \rightarrow aA$
- 2) $S \rightarrow aB$
- 3) $A \rightarrow a$
- 4) $B \rightarrow b$

דוגמא - SLR

• אוטומט פרפיקסי - נשאר אותו דבר:



0) $S' \rightarrow S$

1) $S \rightarrow aA$

2) $S \rightarrow aB$

3) $A \rightarrow a$

4) $B \rightarrow b$

SLR - דוגמא

• בניית טבלת הניתוח:

$\text{follow}(S) = \text{follow}(A) = \text{follow}(B) = \{\$, \}$

LR(0) actions			
	a	b	\$
0	s2		
1			acc
2	s5	s6	
3	r2	r2	r2
4	r1	r1	r1
5	r3	r3	r3
6	r4	r4	r4

goto		
S	A	B
1		
	4	3

0) $S' \rightarrow S$

1) $S \rightarrow aA$

2) $S \rightarrow aB$

3) $A \rightarrow a$

4) $B \rightarrow b$

SLR - דוגמא

• בניית טבלת הניתוח:

$\text{follow}(S) = \text{follow}(A) = \text{follow}(B) = \{\$, \}$

SLR actions			
	a	b	\$
0	s2		
1			acc
2	s5	s6	
3	r2	r2	r2
4	r1	r1	r1
5	r3	r3	r3
6	r4	r4	r4

goto		
S	A	B
1		
	4	3

דוגמא - קונפליקט R\R

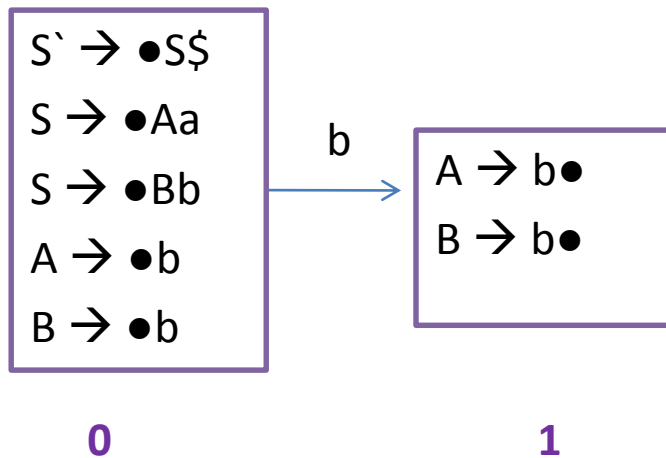
0. $S' \rightarrow S\$$

1. $S \rightarrow Aa$

2. $S \rightarrow Bb$

3. $A \rightarrow b$

4. $B \rightarrow b$

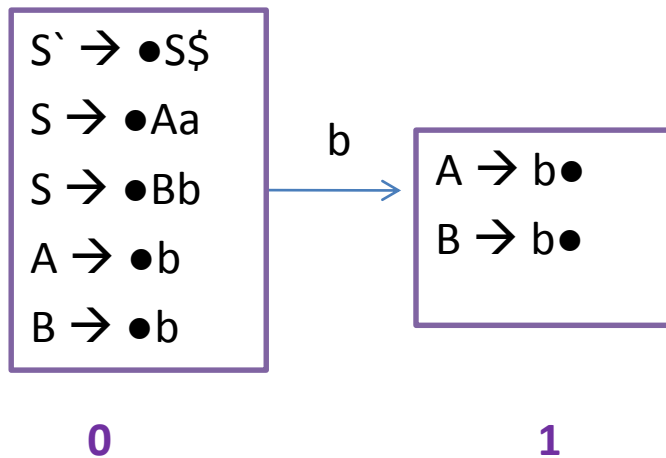


LR(0)

	a	b	\$
0		s1	
1	r3,r4	r3,r4	r3,r4
...			
...			
...			

דוגמא - קונפליקט R\R

- 0. $S' \rightarrow S\$$
- 1. $S \rightarrow Aa$
- 2. $S \rightarrow Bb$
- 3. $A \rightarrow b$
- 4. $B \rightarrow b$



Follow(A)={a}
Follow(B)={b}

SLR

	a	b	\$
0		s1	
1	r3,r4 r3	r3,r4 r4	r3,r4
...			
...			
...			

נשים לב כי מעבר ל SLR פותר את הקונפליקט במקרה זה.

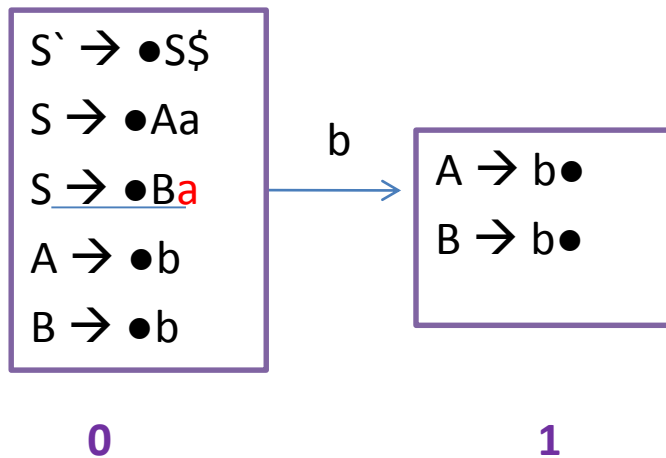
דוגמא - קונפליקט R\R

- אבל עבור הדקדוק הבא, עדיין יש בעיה:

0. $S' \rightarrow S\$$
1. $S \rightarrow Aa$
2. $S \rightarrow Ba$
3. $A \rightarrow b$
4. $B \rightarrow b$

Follow(A)={a}
Follow(B)={a}

SLR



	a	b	\$
0		s1	
1	r3, r4		
...			
...			
...			

בדוגמא זו מעבר ל SLR אינו פותר את הקונפליקט.

Canonical LR (CLR / LR(1))

- אלגוריתם LR(1) מבוסס על מנתח LR בדומה לאלגוריתמים הקודמים שראינו.
- הרעיון:
 - על מנת למנוע קונפליקטים, נפרק את מצבי LR(0) למצבים עדינים יותר, המכילים גם מידע על lookahead
 - עידון של ההקשר: מה מגיע אחרי במסלול ספציפי במקום בכל מקום אפשרי בשפה

$$A \rightarrow \alpha \bullet B \beta \Rightarrow A \rightarrow \alpha \bullet B \beta, t$$

פריטי LR(1)

• ההבדל בין פריטי LR(0) ו-LR(1) נעוץ רק ב-closure.

- פריט LR(0) הוא $(A \rightarrow \alpha \bullet \beta)$ כאשר $A \rightarrow \alpha \beta \in P$
- סגור (closure) על קבוצת פריטים I מוגדר באופן אינדוקטיבי:
 - בסיס: $\text{closure}(I) = I$
 - צעד: אם $(A \rightarrow \alpha \bullet B \beta) \in \text{closure}(I)$ אז לכל $B \rightarrow \gamma \in P$ גם $(B \rightarrow \bullet \gamma) \in \text{closure}(I)$
- פונקצית המעברים של האוטומט:
$$\delta(I, X) = \bigcup \{ \text{closure}(A \rightarrow \alpha X \bullet \beta) \mid (A \rightarrow \alpha \bullet X \beta) \in I \}$$

- פריט LR(1) הוא $(A \rightarrow \alpha \bullet \beta, t)$ כאשר $A \rightarrow \alpha \beta \in P, t \in T \cup \{\$, \}$
- סגור (closure) על קבוצת פריטים I מוגדר באופן אינדוקטיבי:
 - בסיס: $\text{closure}(I) = I$
 - צעד: אם $(A \rightarrow \alpha \bullet B \beta, t) \in \text{closure}(I)$ אז לכל $B \rightarrow \gamma \in P$ ולכל $x \in \text{first}(\beta t)$ גם $(B \rightarrow \bullet \gamma, x) \in \text{closure}(I)$
- פונקצית המעברים של האוטומט:
$$\delta(I, X) = \bigcup \{ \text{closure}(A \rightarrow \alpha X \bullet \beta, t) \mid (A \rightarrow \alpha \bullet X \beta, t) \in I \}$$

משמעות פריט LR(1): זיהינו את מה שמשמאל לנקודה, אנו מצפים למצוא את מה שמימין לה, ולאחר מכן את האסימון המצורף לפריט.

LR(1) – פתרון קונפליקט R\R

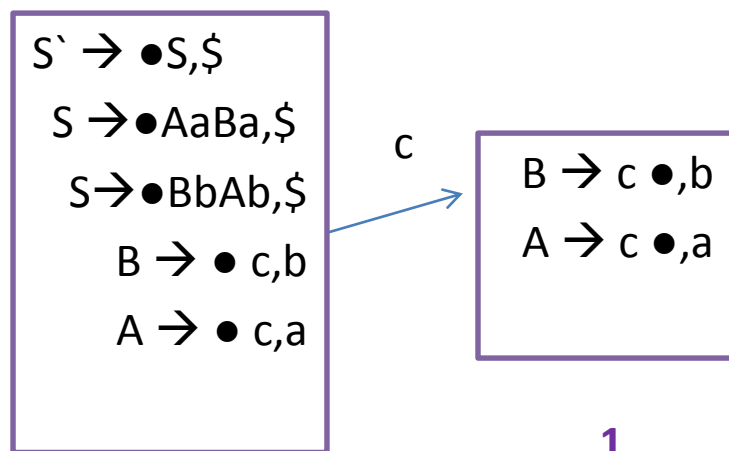
0. $S' \rightarrow S$

1. $S \rightarrow AaBb$

$S \rightarrow BbAb$.2

2. $B \rightarrow c$

3. $A \rightarrow c$



0

1

LR(1)

	a	b	c	\$
0			s1	
1	r3	r2		
...				
...				
...				

בניית טבלת הניתוח

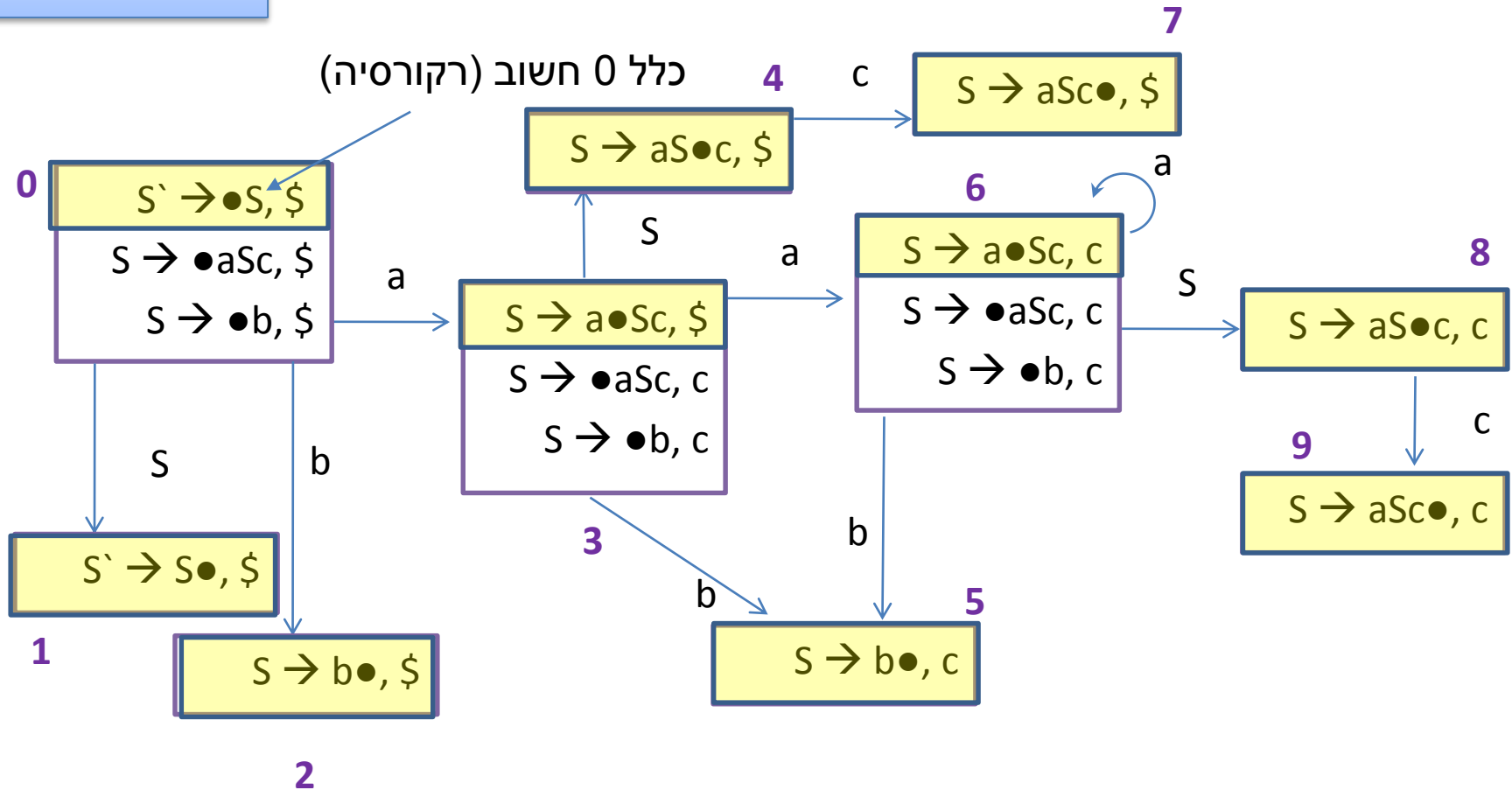
- בונים טבלת Action וטבלת goto כמו LR(0)/SLR

$$\text{action}[i, t] = \begin{cases} \text{SHIFT}_j & \delta(I_i, t) = I_j \\ \text{REDUCE}_k & \text{rule } k \text{ is } A \rightarrow \alpha, \quad (A \rightarrow \alpha \bullet, t) \in I_i \\ \text{ACCEPT} & (S' \rightarrow S \bullet, \$) \in I_i \quad \text{and } t = \$ \\ \text{ERROR} & \text{otherwise} \end{cases}$$

1. $S \rightarrow aSc$

2. $S \rightarrow b$

דוגמא נוספת: רקורסיה



*החלקים בצהוב הם הגרעין של המצב.

0. $S' \rightarrow S$

1. $S \rightarrow aSc$

2. $S \rightarrow b$

דוגמא (1) - LR הרצה

הרצה באופן דומה ל LR(0)

actions					goto
	a	b	c	\$	S
0	s3	s2			1
1				acc	
2				r2	
3	s6	s5			4
4			s7		
5			r2		
6	s6	s5			8
7				r1	
8			s9		
9			r1		

מחסנית

פעולה

קלט

(0,)

s3

aabcc\$

(0,)(3,a)

s6

abcc\$

(0,)(3,a)(6,a)

s5

bcc\$

(0,)(3,a)(6,a)(5,b)

r2

cc\$

(0,)(3,a)(6,a)(goto(6,S),S)

(0,)(3,a)(6,a)(8,S)

s9

cc\$

(0,)(3,a)(6,a)(8,S)(9,c)

r1

c\$

(0,)(3,a)(goto(3,S),S)

(0,)(3,a)(4,S)

s7

c\$

(0,)(3,a)(4,S)(7,c)

r1

\$

(0,)(goto(0,S),S)

(0,)(1,S)

acc

\$

LR סיכום

- מנתחי LR בונים את תת-העץ האפשרי הראשון במעבר על הקלט
- קונפליקטים נוצרים כאשר
 - יש יותר מתת-עץ אפשרי אחד
 - אפשר לבנות תת-עץ או להמשיך לקרוא קלט
- כדי לאפשר יותר דקדוקים, דרוש יותר הקשר
- הקשר "עולה" בזמן חישוב (follow) ובזיכרון (גודל האוטומט והטבלה)

שאלה 1

נתון $G=(V, T, P, S)$ השייך ל $LR(1)$. עבור כל אחד מהדקדוקים הבאים, רשמו האם הוא **בהכרח** ב $LR(1)$:

א. $G_1 = (V \cup \{X\}, T, P \cup \{X \rightarrow t\}, S)$ כאשר $t \in T, X \notin V$

ב. $G_2 = (V, T, P \cup \{S \rightarrow \varepsilon\}, S)$

שאלה 1

כאשר $t \in T, X \notin V$

$G_1 = (V \cup \{X\}, T, P \cup \{X \rightarrow t\}, S)$.א

• הדקדוק בהכרח ב LR(1):

– המשתנה X הוא משתנה חדש שאינו נגזר באגף ימין של אף כלל.

– בזמן ביצוע closure במצבי האוטומט, הכלל $X \rightarrow t$ לעולם לא ייכנס.

– האוטומט יהיה זהה לאוטומט של G , שהוא חסר קונפליקטים.

X משתנה לא ישיג !

שאלה 1

$$\text{ב. } G_2 = (V, T, P \cup \{S \rightarrow \varepsilon\}, S)$$

• אינו בהכרח ב LR(1).

– הכלל החדש יכול להפוך את הדקדוק לדו משמעי.
לדוגמא:

$$G = \{S \rightarrow A, A \rightarrow \varepsilon\}$$

$$G_2 = \{S \rightarrow A, A \rightarrow \varepsilon, S \rightarrow \varepsilon\}$$

דקדוק דו משמעי: יש שתי גזירות ימניות ביותר בדקדוק לאותה מילה.

מספר דגשים

- להוכחה ששפה מסויימת היא LR צריך לצייר את כל האוטומט ולהראות כי אין קונפליקטים באף מצב

- להפרכה, מספיק מסלול אחד באוטומט

שאלה ממבחן

- נתונה שפת סקריפטינג המורכבת ממשתנים גלובליים ורשימת פונקציות ספריה שלא מקבלות פרמטרים.
- תכנית מורכבת משורה אחת של אפס או יותר קריאות פונקציה על משתנה. מותר לקרוא לכל פונקציה על כל משתנה גלובאלי ועל כל תוצאה של הפעלת פונקציה. תכנית בשפה תיראה כך:
`grades.sort().top5().awardExcellence()`
- כבר מומש עבורכם ניתוח לקסיקוגרפי המחזיר את האסימונים id (שם משתנה או פונק'), dot (נקודה) ו-pars (פתח וסגור סוגריים).

א. בנו דקדוק לשפה כך שניתן יהיה לבנות לו מנתח
מאחת המחלקות שנלמדו בקורס והוכיחו כי ניתן.

$$\begin{aligned} \textit{Program} &\rightarrow \textit{id Funcs} \\ \textit{Funcs} &\rightarrow \textit{dot id pars Funcs} \\ \textit{Funcs} &\rightarrow \epsilon \end{aligned}$$

באיזו מחלקה הדקדוק? (פתרון על הלוח)