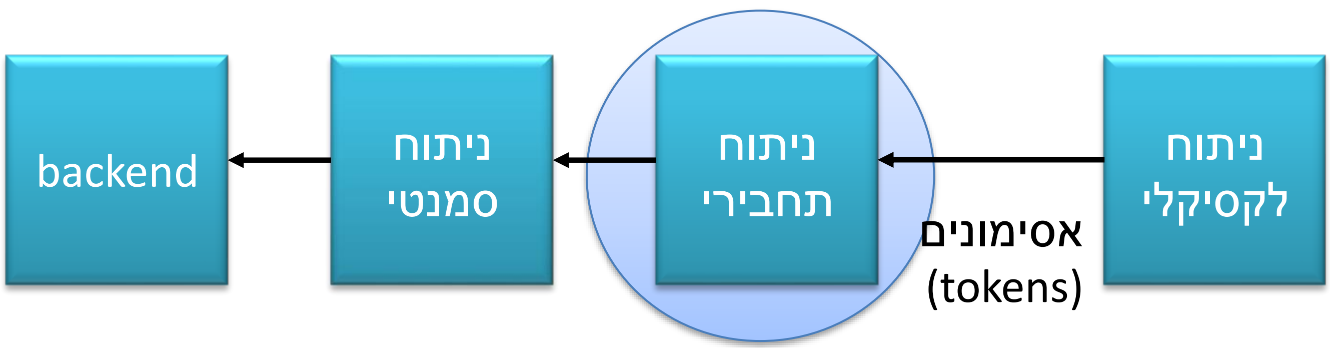
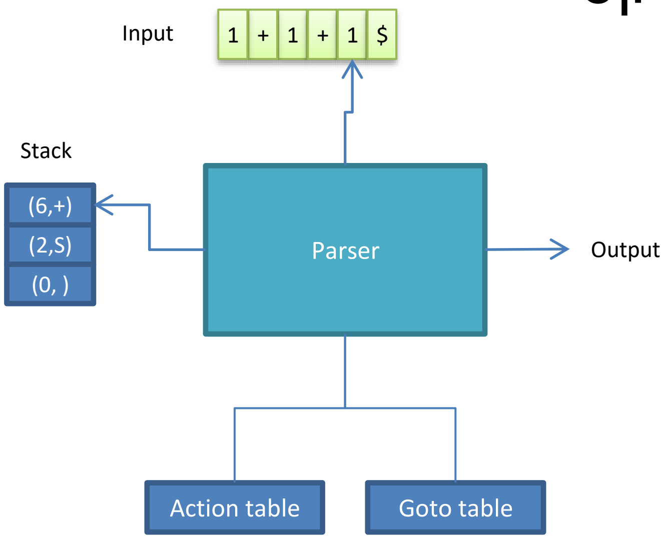
תזכורת מתרגולים אחרונים

- מבנה סכמתי של קומפיילר



- ניתוח תחבירי:
  - **Top Down**: LL(1)
  - **Bottom up**:
    - LR(0)
    - SLR
    - LR(1)

בניית מנתח LR



<u>שלבי בניית המנתח:</u>

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

<u>הרצת המנתח על קלט:</u>

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.

- <u>פריט LR(0)</u> הוא (A→α●β) כאשר $A \rightarrow \alpha\beta \in P$
  - כל אחד ממצבי אוטומט 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 → ε ?

מנתח LR(0) – בניית האוטומט

- <u>פריט LR(0)</u> הוא (A→α●β) כאשר $A \to \alpha\beta \in P$
  - כל אחד ממצבי אוטומט המנתח הוא קבוצת פריטי LR(0).

- <u>סגור (closure)</u> על קבוצת פריטים I מוגדר באופן אינדוקטיבי:
  - בסיס: closure(I)=I
  - צעד: אם (A→α●Bβ) ∈closure(I) אז

    לכל B→γ ∈ P גם (B→●γ) ∈closure(I)

קבוצת פריטי LR(0)

$X \in T \cup V$

- <u>פונקצית המעברים של האוטומט ממצב</u> I <u>...ימן</u> X:

$$\delta(I, X) = \bigcup \left\{ closure\left(A \to \alpha X \bullet \beta\right) \middle| \left(A \to \alpha \bullet X \beta\right) \in I \right\}$$

מנתח LR(0) – בניית האוטומט

לפני בניית האוטומט מוודאים שיש כלל התחלתי בודד ולא רקורסיבי ובמידה ואין אז נוסיף $S' \rightarrow \cdot S$.
ניתן תמיד להוסיף את הכלל ולא לחשוב על זה (אלא אם נאמר אחרת).

אלגוריתם בנית האוטומט

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

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

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) $B \rightarrow b$

LR (0) - דוגמא

- <u>בניית אוטומט פרפיקסי:</u>

0

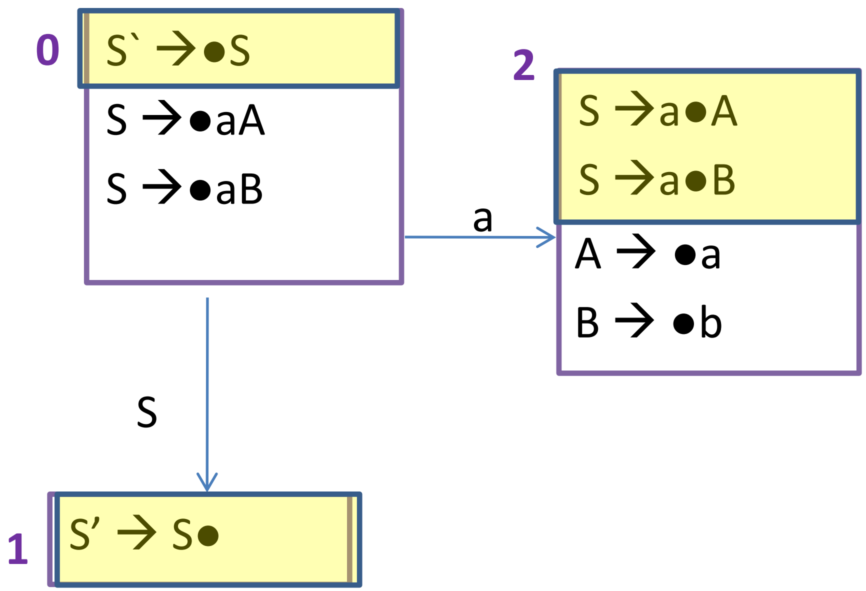
$S` \rightarrow \bullet S$
$S \rightarrow \bullet aA$
$S \rightarrow \bullet aB$

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

0) S’ → S
1) S → aA
2) S → aB
3) A → a
4) B → b

LR (0) - דוגמא

• <u>בניית אוטומט פרפיקס:</u>



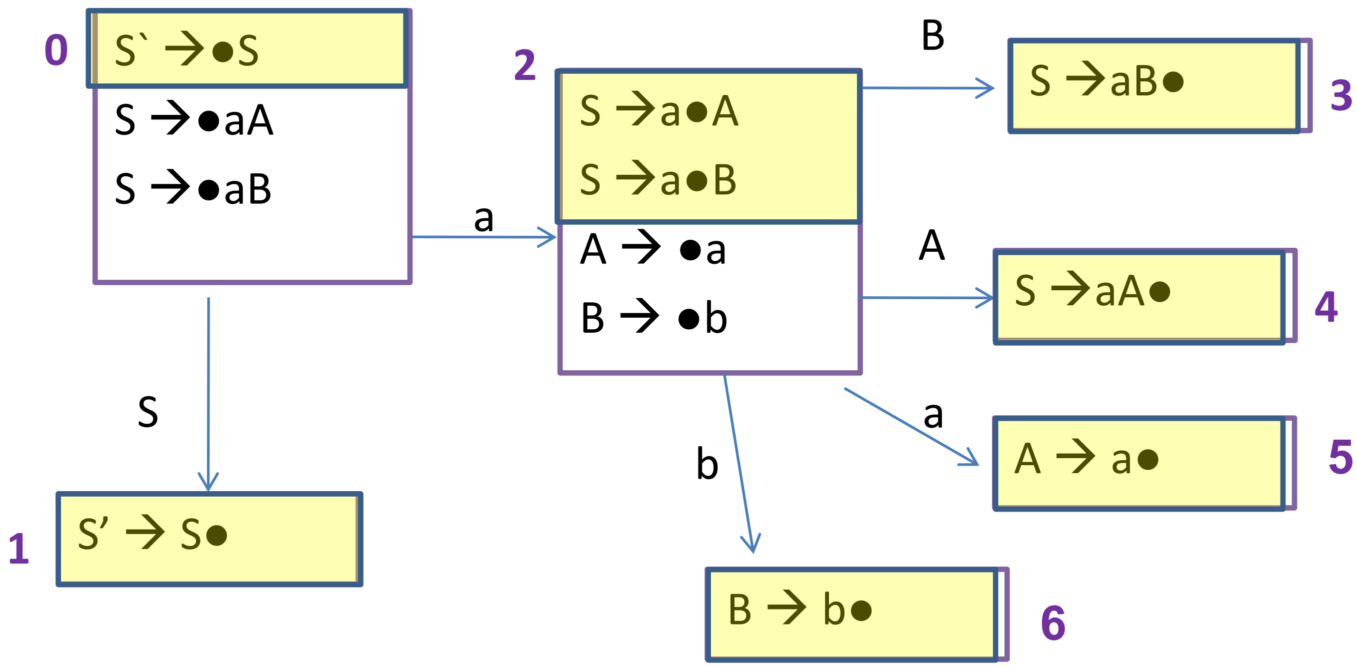
1. לכל סימן $X \in V \cup T$ עבורו קיים פריט $A \to \alpha \cdot X\beta \in I$:
   1. מחשבים את $\delta(I,X)$ (ומוסיפים אותו לקב' המצבים, אם עדיין לא חלק ממנה)
   2. יוצרים קשת עם הסימן X, שמובילה למצב $\delta(I,X)$.

$$\delta(I,X) = \bigcup \left\{ closure(A \to \alpha X \bullet \beta) \middle| (A \to \alpha \bullet X\beta) \in I \right\}$$

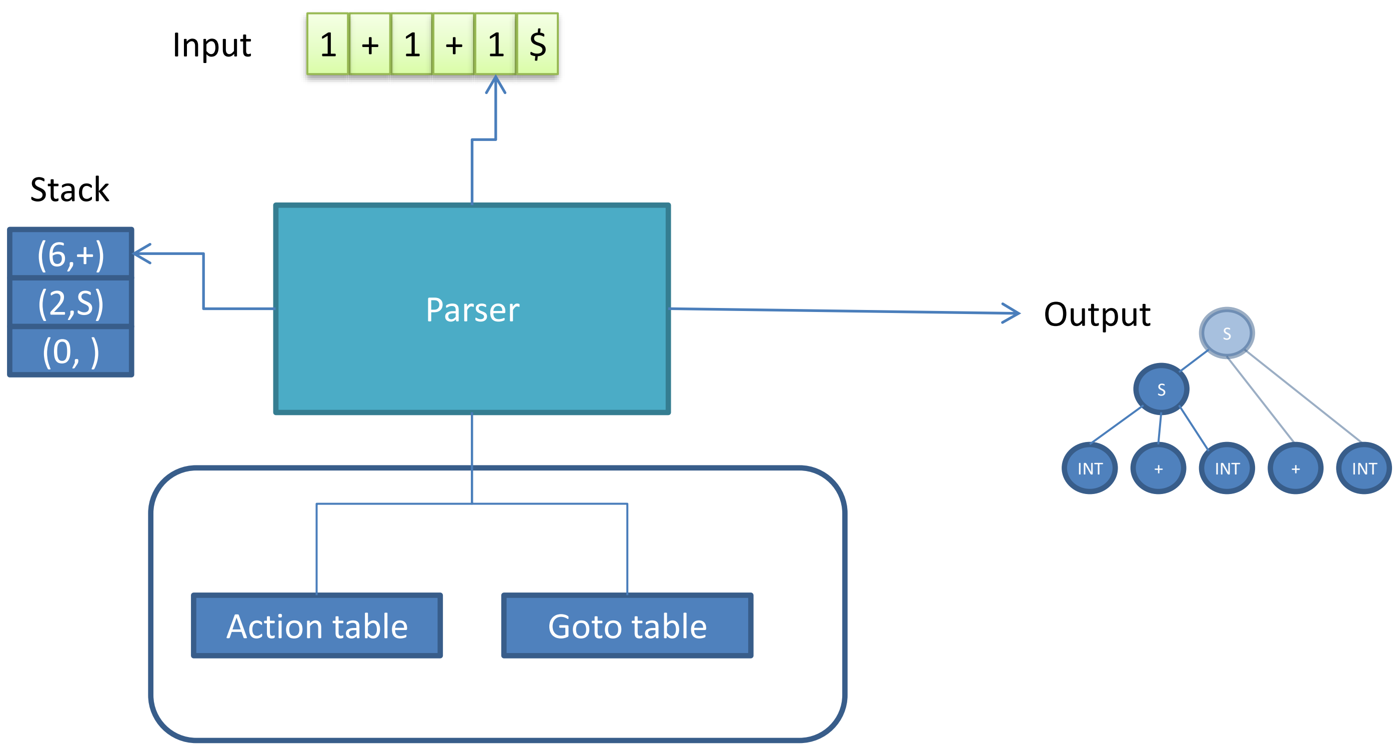
0) S’ → S
1) S → aA
2) S → aB
3) A → a
4) B → b

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

$t \in T$

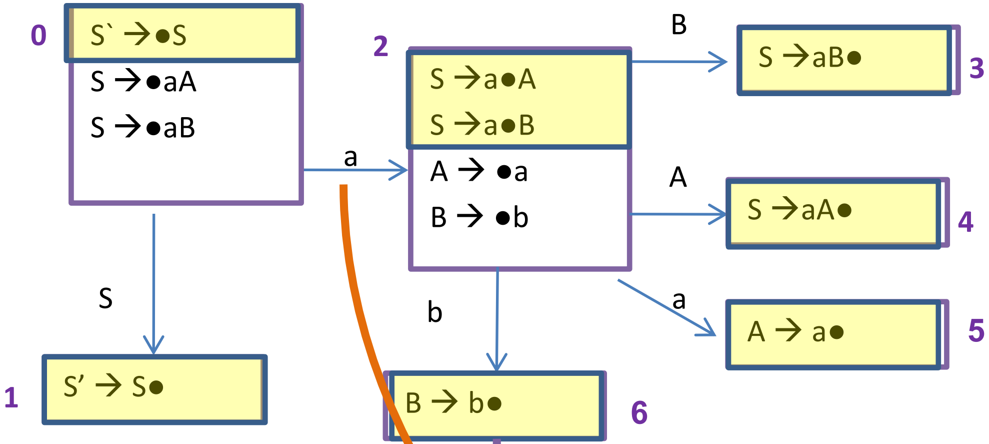
$$\text{action}[i,t] = \begin{cases} \text{SHIFT}_j & \delta(I_i,t) = I_j \\ \text{REDUCE}_k & \text{rule k is } A \rightarrow \alpha, (A \rightarrow \alpha \bullet) \in I_i \\ \text{ACCEPT} & (S' \rightarrow S \bullet) \in I \text{ and } t = \$ \\ \text{ERROR} & \text{otherwise} \end{cases}$$

הגדרת טבלת goto למנתח LR(0):

$X \in V$

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

דוגמא –
בניית טבלת הניתוח



0) S' → S
1) S → aA
2) S → aB
3) A → a
4) B → b

מספר המצב אליו עוברים

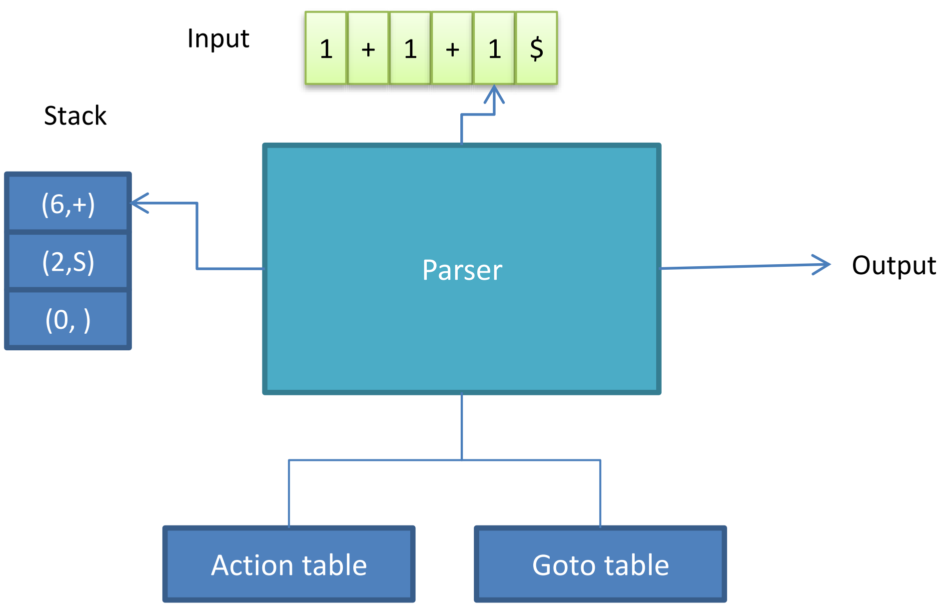
	actions			goto		
	a	b	$	S	A	B
0	s2			1		
1			acc			
2	s5	s6			4	3
3	r2	r2	r2			
4	r1	r1	r1			
5	r3	r3	r3			
6	r4	r4	r4			

מספר הכלל שגוזרים

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’ → S
1) S → aA
2) S → aB
3) A → a
4) B → b

דוגמא LR (0) - הרצה

רצף האסימונים בקלט: ab$

Action

	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
0	1		
1			
2			3
3			
4			
5			
6			

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,t).
  2. קדם את הראש הקורא את הקלט צעד אחד ימינה.

0) S’ → S
1) S → aA
2) S → aB
3) A → a
4) B → b

	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

דוגמא LR (0) - הרצה

רצף האסימונים בקלט: ab$

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→α, כלל הגזירה שמספרו הוא j):
  1. הוצא |α| 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’ → S
1) S → aA
2) S → aB
3) A → a
4) B → b

דוגמא (0) LR - הרצה

רצף האסימונים בקלט: ab$

Action

	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
0	1		
1			
2			3
3			
4			
5			
6			

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 :

$A \rightarrow \alpha \bullet t\beta$
$B \rightarrow \delta \bullet$

האם להמשיך לקרוא את הקלט או לצמצם ?

2) קונפליקט reduce/reduce :

$A \rightarrow \alpha \bullet$
$B \rightarrow \delta \bullet$

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:

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

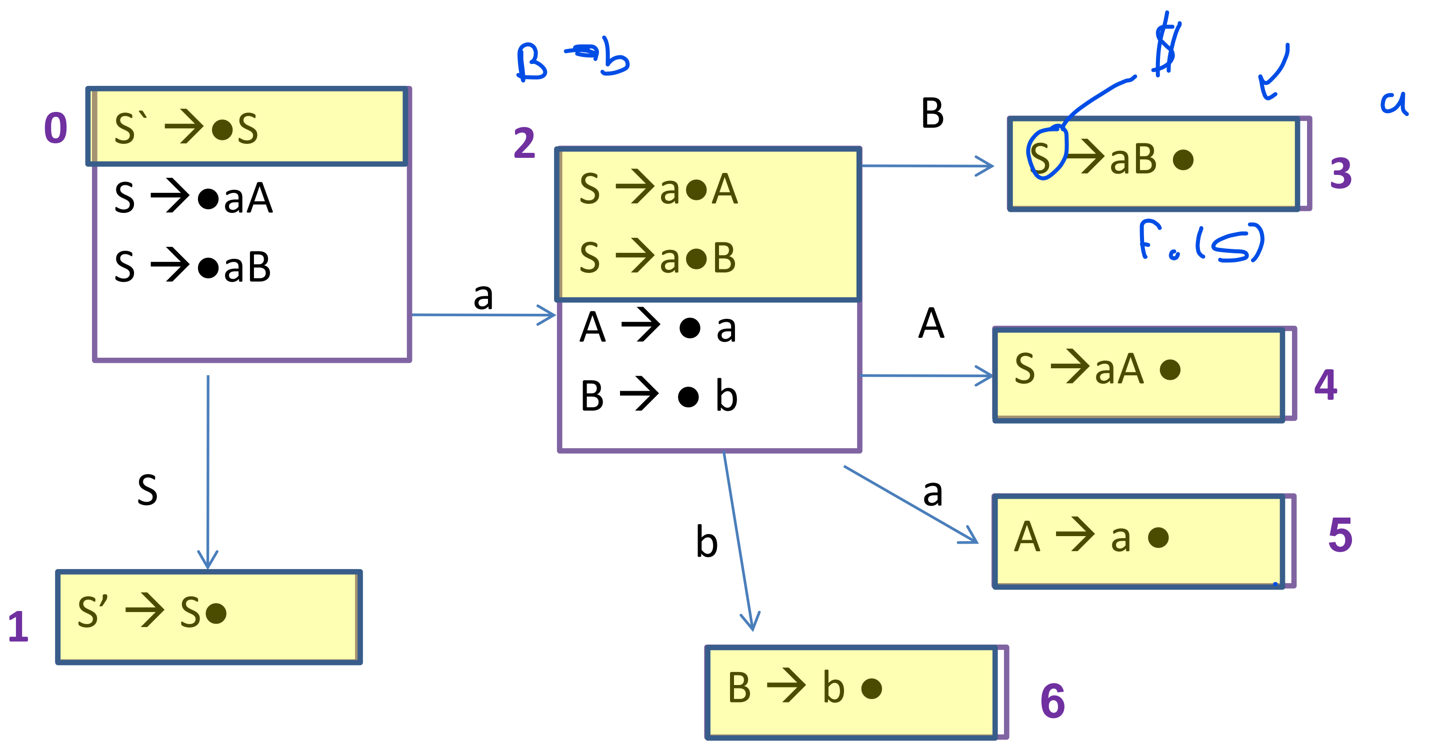
0) S’ → S
1) S → aA
2) S → aB
3) A → a
4) B → b

SLR - דוגמא

S' → aSBc

S → ε

- אוטומט פרפיקסי – נשאר אותו דבר:



0) S' → S
1) S → aA
2) S → aB
3) A → a
4) B → b

דוגמא - SLR

• בניית טבלת הניתוח:

follow(S) = follow(A) = follow(B) = {$}

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

0) S' → S
1) S → aA
2) S → aB
3) A → a
4) B → b

SLR - דוגמא

- <u>בניית טבלת הניתוח:</u>

follow(S) = follow(A) = follow(B) = {$}

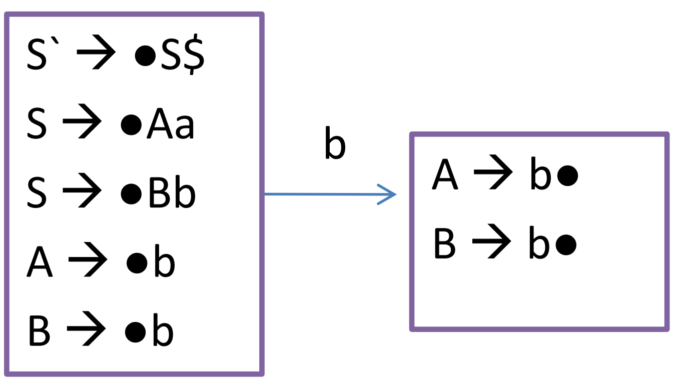
SLR actions			
	a	b	$
0	s2		
1			acc
2	s5	s6	
3	~~r2~~	~~r2~~	r2
4	~~r1~~	~~r1~~	r1
5	~~r3~~	~~r3~~	r3
6	~~r4~~	~~r4~~	r4

goto		
S	A	B
1		
	4	3

דוגמא - קונפליקט R\R

0. S` → S$
1. S → Aa
2. S → Bb
3. A → b
4. B → b

LR(0)

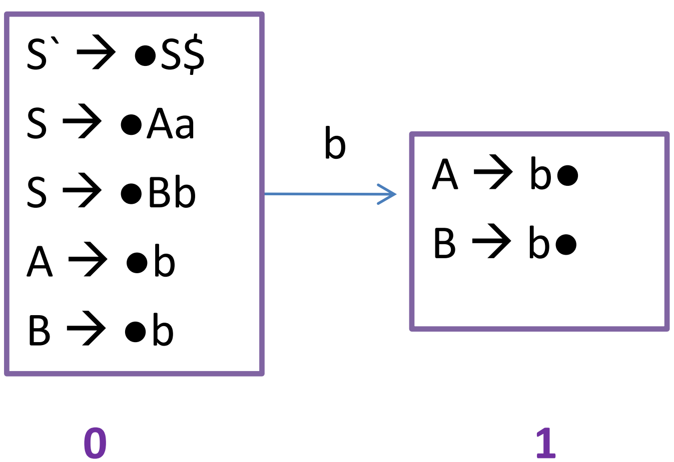


0 **1**

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

דוגמא - קונפליקט R\R

0. S` → S$
1. S → Aa
2. S → Bb
3. A → b
4. B → b



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

SLR

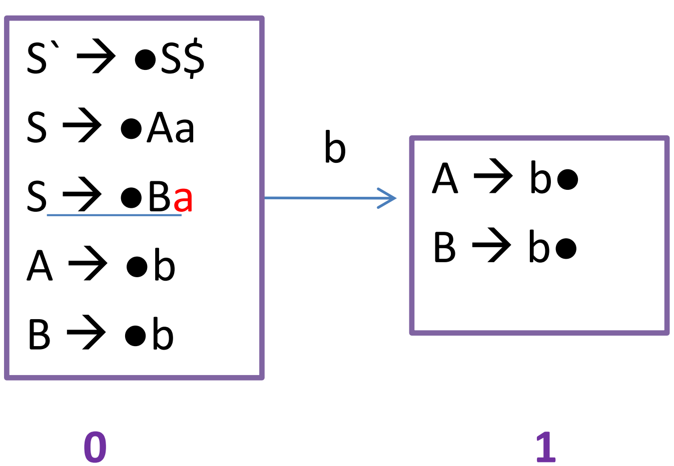
	a	b	$
0		s1	
1	~~r3,r4~~ r3	~~r3,r4~~ r4	~~r3,r4~~
…			
…			
…			

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

דוגמא - קונפליקט R\R

- אבל עבור הדקדוק הבא, עדיין יש בעיה:

0. S` → S$
1. S → Aa
2. S → Ba
3. A → b
4. B → b



SLR

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

	a	b	$
0		s1	
1	r3, r4		
…			
…			
…			

בדוגמא זו מעבר ל SLR אינו פותר את הקונפליקט.

נפתור
בהרצאה\תרגול
הבאים

מספר דגשים

- להוכחה שדקדוק מסויים הוא LR צריך לצייר את **כל** האוטומט ולבנות את הטבלה כדי להראות 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.

$$Program \rightarrow id\ Funcs$$
$$Funcs \rightarrow dot\ id\ pars\ Funcs$$
$$Funcs \rightarrow \epsilon$$

באיזו מחלקה הדקדוק? (פתרון על הלוח)