תורת הקומפילציה – סיכום

לקסמה (Lexeme) – תת-מחרוזת בקלט אשר מתאימה לאסימון מסוים. למשל אם יש אסימון של integer אז 242 יהיה לקסמה מתאימה לאסימון.

אסימון (Token) – יחידה אשר משמשת כטרמינל בודד עבור המנתח התחבירי. לדוגמא Int או "=".

נכונות אלגוריתם Top Down – אלגוריתם ניתוח Top Down הוא נכון אם הוא מקיים:

א. נאותות (soundness): עבור כל מילה שאינה שייכת לשפה הוא מודיע על שגיאה

ב. שלמות (completeness): עבור כל מילה השייכת לשפה הוא מוצא עץ גזירה

יעילות אלגוריתם Top Down – אלגוריתם ניתוח Top-Down הוא יעיל (efficient) אם עבור כל מילה שהוא מוצא עבורה עץ גזירה, הוא מוצא אותו ללא backtracking (או ללא ניחוש).

פונקציית first: שייכת לכלל גזירה מסוים. כוללת קבוצה של כל הטרמינלים אשר עשויים להופיע בתור האות הראשונה של תת המחרוזת הסופית שתיגזר מהכלל. אם ε נגזרת מהכלל, אזי גם ε שייכת ל first של הכלל.

חישוב פונקציית first עבור כלל $V \rightarrow \alpha$:

אם α מתחילה בטרמינל t אזי ה first(V)=t.

אם α מתחילה במשתנה Y בלתי אפיס (שאינו נגזר ל ε) אזי ה first(V)=first(Y)

אם α מתחילה במשתנה Y_1Y_2 ו- Y_1 אפיס, אז יש לצרף גם את $first(Y_2)$ לקבוצה (וכך הלאה).

פונקציית follow: פונקציה הפועלת על משתני גזירה. Follow(A) = קבוצת כל הטרמינלים אשר יכולים להופיע בצמוד ל-A מימין בנגזרת כלשהי. כלומר $a \in follow(A) \Rightarrow S \overset{*}{\rightarrow} \alpha A a \beta$.

הערה: גם -| יכול להיות חלק מ follow(A), והוא תמיד חלק מ follow של המשתנה ההתחלתי.

חישוב פונקציית follow(A):

אתחול: מאתחלים קבוצה ריקה בשם follow עבור כל משתנה $X \in V$.

מוסיפים את סימן סוף הקלט ($\dashv$) לקבוצת ה-follow של המשתנה ההתחלתי (S).

צעד: מבצעים את הפעולה הבאה על כל המשתנים (X) עד שאף קבוצת follow אינה משנה את ערכה:

אם X מופיע באגף ימני של כלל גזירה (מהצורה $Y \rightarrow \alpha X \beta$), אז:

(בכל מקרה) $$follow(X) = follow(X) \cup first(\beta)$$

ואם $\beta \overset{*}{\Rightarrow} \varepsilon$, אז: $$follow(X) = follow(X) \cup follow(Y)$$

ערך הפונקציה follow הוא הערך הסופי של קבוצות ה-follow המתקבלות.

פונקציית select: פונקציה הפועלת על כללי גזירה. קובעת עבור אילו טרמינלים בקלט יש להשתמש בכלל גזירה

חישוב select:
$$select(A \rightarrow \alpha) = \begin{cases} first(\alpha) \cup follow(A) & \alpha \overset{*}{\Rightarrow} \varepsilon \\ first(\alpha) & otherwise \end{cases}$$

דקדוק LL(1): דקדוק G יקרא דקדוק LL(1) אם"ם לכל שני חוקים ב- G השייכים לאותו משתנה A מתקיים:

$$select(A \rightarrow \alpha) \cap select(A \rightarrow \beta) = \phi$$

(כלומר, בטבלת הניתוח קיים לכל היותר חוק אחד בכל משבצת)

תזכורות:

1. *זהו בדיוק התנאי הנדרש עבור בנית מנתח RD-משופר.*
2. *בהחלט יתכן ששני דקדוקים ח"ה יגזרו אותה שפה, כך שאחד מהם הוא LL(1), והשני איננו LL(1).*
3. *משמעות השם: (Left to right scan, Leftmost derivation, 1 lookahead symbol) – קריאה משמאל לימין, גזירה שמאלית ביותר, וראיית סמל אחד בלבד קדימה.*
4. *לא כל דקדוק ניתן להביא למצב LL(1).*

ניתוח דקדוק LL(1):

הכנה:
1. חשב select עבור כל כלל גזירה. (לצורך זה חשב first ו- follow).
2. בנה את טבלת הניתוח M להלן:
   a. ב-M יש עמודה לכל טרמינל (כולל |-), ושורה לכל משתנה.
   b. במשבצת M[V,t] (כלומר בשורה של המשתנה V בעמודה של הטרמינל t) שים את מספרו של כל כלל גזירה מהצורה $V \rightarrow \alpha$ אשר מקיים $t \in select(V \rightarrow \alpha)$.
   c. אם במשבצת מסויימת יש יותר מכלל גזירה אחד, אזי הדקדוק אינו LL(1) ויש לצאת בשגיאה.

הניתוח יתבצע בטבלה הבאה (עם שורה לדוגמא):

פעולה	הטרמינל בראש הקלט	מצב המחסנית הנוכחי
replace (1)	assign	$ P

על המחסנית מסתכלים מימין לשמאל.

האתחול מתבצע ע"י דחיפת הסימן $ למחסנית (או |-), דחיפת המשתנה התחילי למחסנית, ואתחול מצביע הקלט לאות הראשונה של הקלט.

בכל שורה בודקים את האות שבראש המחסנית.
<u>אם היא טרמינל</u>: בודקים אם היא תואמת את ראש הקלט. אם לא – שגיאה. אם כן – עושים Shift, מסירים את הטרמינל מהמחסנית, ומזיזים את מצביע הקלט באחד.
אם היא $

<u>אם היא משתנה</u>: שלוף את המשתנה מהמחסנית, והכנס את **<u>ההיפוך</u>** של כלל הגזירה המתאים (נניח j) לאותו משתנה (ע"פ טבלת הניתוח) למחסנית. כתוב בפעולה replace(j).

<u>ניתוח LR(k)</u>:

משמעות השם (Left to right scan, Rightmost derivation, k lookahead symbols). כאשר משמיטים את k, אזי k=1.
- מנתחי LR הם יעילים מאוד (סיבוכיות לינארית יחסית לאורך הקלט)
- מנתחי LR מגלים שגיאות תחביריות בהזדמנות הראשונה האפשרית לגילויין במסגרת של קריאת קלט משמאל לימין.
- מוצאים את הגזירה הימנית (R) ביותר בסדר הפוך לסדר הגזירה. (חשוב עבור פעולות סמנטיות, ובוודאי עבור דקדוקים רב-משמעיים)
- אוסף הדקדוקים הניתנים לפריסה בשיטה זו מכיל ממש את אוסף הדקדוקים הניתנים לפריסה על ידי פורס מנבא

<u>בניית מנתח LR(1) בשיטת SLR(1):</u>

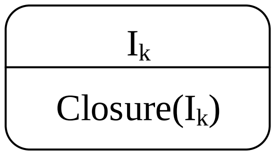
א. בניית האוטומט הפרפיקסי של מנתח SLR(1).
בתור הכנה יש לחשב את ה follow של כל משתנה.
<u>פריט:</u> (item) של דקדוק G הוא חוק של G המכיל נקודה במיקום מסויים באגף הימני של החוק.

<u>סגור:</u> (closure) של קבוצת פריטים I מוגדר אינדוקטיבית באופן הבא:
1. כל הפריטים ב- I נמצאים גם ב- Closure(I).
2. אם הפריט: $A \rightarrow \alpha \cdot B\gamma$ נמצא ב- Closure(I), אז כל פריט מהצורה: $B \rightarrow \cdot\delta$ גם ב- Closure(I).
   (כאשר B הוא משתנה, ו- $\alpha,\gamma,\delta \in (V \cup T)^*$)
   אם יש כלל מהצורה $B \rightarrow \varepsilon$, אז נוסיף את הפריטר: $B \rightarrow \cdot$
   הערה: כלל 2 מופעל רק אם יש סימנים מימין לנקודה.

<u>פונקציית דלתא (או goto בספרים)</u>: מופעלת על קבוצת פריטים I, וסימן X (משתנה או טרמינל).
פורמלית: $\delta(I,X) = \bigcup_{A\rightarrow\alpha.X\beta\in I} Closure(\{A \rightarrow \alpha X.\beta\})$ (יש לשים לב שהנקודה משמאל ל-X ב-I ואילו הסגורים מחושבים על הפריטים בהם הנקודה מימין ל-X).

יש להוסיף חוק לדקדוק מהצורה $S' \rightarrow S$ (0).

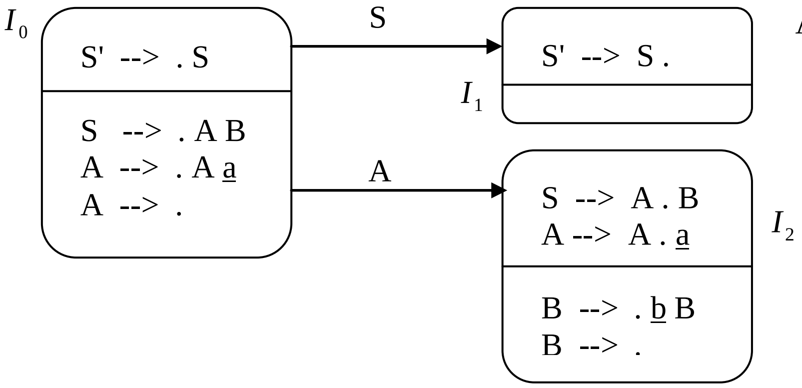
כל מצב באוטומט יראה כך:



אתחול: I_0 הוא $S` \rightarrow .S$.

פיתוח עבור כל I_k שעוד לא פותח: חשב את איחוד הסגורים של כל הפריטים ב- I_k ($Closure(I_k)$), ועבור כל סימן X (טרמינל או משתנה) אשר מופיע מיידית מימין לנקודה בפריט כלשהו במצב, צור חץ למצב חדש אשר יכיל את $\delta(I_k, X)$.

ניתן לראות בדוגמא כי במצב I_0 הופיעו הסימנים S, ו- A מימין לנקודה בפריטים. לכן יצא חץ של S למצב I_1 אשר כלל את כל הפריטים ב $\delta(I_k, X)$.



<u>בניית טבלת הניתוח:</u>

בנויה משתי תתי טבלאות: טבלת Action, ו goto.

נראית כך:

state	action				Goto			
	t_1	t_2	t_3	⊣	**S**	v_1	v_2	v_3
0								
1								
2								

כלומר, בטבלת Action יש עמודה לכל טרמינל, ול -| , ובטבלת goto יש עמודה לכל משתנה. השורות בטבלה מתאימות למצבי האוטומט.

עבור כל מצב עבור על כל החיצים שיוצאים ממנו:

1. אם החץ שייך לטרמינל, ומצביע למצב I_j, רשום בעמודה של הטרמינל, בשורה של המצב S_j (כלומר shift למצב j).
2. אם החץ שייך למשתנה, ומצביע למצב I_j רשום בטבלת ה goto בעמודה של המשתנה בשורה של המצב J.

אם המצב כולל פריט סופי כלשהו (פריט אשר מסתיים בנקודה. כלומר: $V \rightarrow \alpha \bullet$) אשר שייך לכלל גזירה מספר k:

רשום R_k בעמודה של כל טרמינל השייך ל follow(V).

אם יש במשבצת אחת יותר מפעולה אחת, יש קונפליקט והדקדוק אינו SLR(1).

<u>אלגוריתם הניתוח עצמו:</u>

האלגוריתם כללי לכל דקדוק LR. רק אופן בניית טבלת הניתוח הוא ספציפי לשיטות שונות (כמו CLR, LALR, SLR).

<u>אתחול:</u> מאתחלים מחסנית ריקה, ודוחפים אליה את המצב ההתחלתי (0).

<u>צעד:</u>

1. בודקים את המצב בקצה המחסנית (נניח n), ואת התו הבא בקלט (נניח t).
2. אם הפעולה בשורה n, בעמודה של t היא S_k אזי:
   א. דוחפים t למחסנית.
   ב. דוחפים k למחסנית.
   ג. מקדמים את הקלט ב-1.
3. אם הפעולה ב בשורה n, בעמודה של t היא R_k (כאשר k הוא כלל גזירה $V \rightarrow \alpha$.) אזי:
   א. מוציאים מהמחסנית $2|\alpha|$ איברים. (כעת המצב בראש המחסנית הוא k')
   ב. דוחפים למחסנית את V.
   ג. דוחפים למחסנית את goto[k',V].
4. אם הפעולה היא accept או error, <u>עוצרים</u>. אחרת, חוזרים ל- 1.

<u>טבלת הרצה עם שורה לדוגמא:</u>

צעד	**מצב המחסנית**	**סרט הקלט**	**פעולה מתבצעת**	**חוק מופעל**
0	0	abb⊣	R_3	A --> ε

מעבר על עמודת "חוק מופעל" מהסוף להתחלה תתן לבסוף את הגזירה הימנית ביותר של המילה

הגדרה מונחית תחביר

תכונה: טיפוס (כלשהו) מוגדר מראש, בעל שם מזהה, הצמוד למשתנה או טרמינל דקדוק מסויים.
שימו לב: תכונה היא טיפוס (דבר מופשט) ולא ערך.
דוגמה: למשתנה A נצמיד תכונה מטיפוס int בשם VALUE, ותכונה מטיפוס boolean בשם EVEN.

מופע של תכונה: התכונה של מופע כלשהו של משתנה או טרמינל בעץ גזירה נתון.
סימון: מופע התכונה VALUE של משתנה A יסומן ע"י A.VALUE.
מופע התכונה EVEN של משתנה A יסומן ע"י A.EVEN.

כלל סמנטי: פקודה המותאמת לכלל גזירה מסויים, המגדירה ערך מופע תכונה ע"י ערכי מופעים אחרים של התכונה או של תכונות אחרות.

תכונות נוצרות: (synthesized) תכונה PROP של משתנה V תקרא נוצרת אם כל החישובים שלה צמודים לכללי גזירה מהצורה: $V \rightarrow \alpha$. (כלומר, התכונה PROP תלויה אך ורק בערכי התכונות של בנים של V בעץ הגזירה הנוצר.)
שימוש: כאשר רוצים להביע תכונות "עצמאיות" (חסרות הקשר) של משתנה V.

תכונות נורשות: (inherited) תכונה PROP של משתנה V תקרא נורשת אם כל החישובים שלה צמודים לכללי גזירה מהצורה: $U \rightarrow \alpha V \beta$. (כלומר, התכונה PROP תלויה אך ורק בערכי התכונות של האחים או האב של V בעץ הגזירה הנוצר.)
שימוש: כאשר רוצים להביע תכונות תלויות-הקשר של משתנה V (כלומר, תכונות התלויות בתכונות שכניו).

הערה: קיימות גם תכונות שאינן נוצרות ואינן נורשות!

טבלאות סמלים:

לצורך בדיקת תחומי הגדרה של משתנים, נצטרך להחזיק מבנה נתונים בצורת עץ דו כיווני (פוינטרים מהאב לבן ומהבן לאב).
כל צומת בעץ תהווה טבלת סמלים עבור שגרה מסוימת הבנים של כל צומת יהיו השגרות אשר מוגדרות בתוך השגרה האב