פישוט דקדוקים

הרעיון הכללי – להגביל את סוגי החוקים שבדקדוק חסר הקשר על מנת להקל על הניתוח של הדקדוקים, אך מבלי לאבד את כוח ההבעה (כל שפה שיש עבורה דקדוק חסר הקשר, יש עבורה דקדוק חסר הקשר פשוט).

- הפישוט לא בהכרח מקטין את מספר כללי הגזירה בדקדוק.

הפעולות מהן מורכב פישוט דקדוק :

1) זריקת משתנים מיותרים
1. משתנים שאינם ניתנים לגזירה טרמינלית.
2. משתנים שלא ניתנים להשגה מ- S.

2) ביטול חוקי- ε : חוקים מהצורה $A\rightarrow\varepsilon$, $A\in V$

3) ביטול חוקי יחידה : חוקים מהצורה $A\rightarrow B$, $A,B\in V$

☜ הסדר שבו יש לבצע את הפעולות מופיע בדפים המצורפים.

סילוק משתנים מיותרים :

- משתנה B אינו ניתן להשגה מ- S אם לא קיימת גזירה $S\overset{*}{\Rightarrow}\alpha B\beta$, $\alpha,\beta\in(V\cup T)^*$
- משתנה B אינו ניתן לגזירה טרמינלית אם לא קיימת גזירה $B\overset{*}{\Rightarrow} w$ עבור $w\in T^*$ כלשהי.
- שני סוגי המשתנים הנ״ל נחשבים מיותרים, שכן השפה של דקדוק היא אוסף כל המילים הטרמינליות שניתן לגזור מהמשתנה ההתחלתי S.

אם מבטלים את כל החוקים של G שבהם מופיעים משתנים מיותרים, מקבלים דקדוק G' כך ש- $L(G) = L(G')$.

איך מוצאים את המשתנים המיותרים ?

1) אלג׳ למציאת המשתנים <u>שניתן</u> לגזור מהם מילה טרמינלית :

useful $\leftarrow \emptyset$
$v(1) = \{A \in V \mid A \rightarrow w \in P, w \in T^*\}$
k=1
while $v(k) \neq \emptyset$ {
  useful $\leftarrow$ useful $\cup$ v(k)
  k++
  $v(k)=\{A \in V \mid A \notin useful\ , A \rightarrow \alpha\ , \alpha \in (useful \cup T)^* \}$
}

כל משתנה שלא מופיע לבסוף ב- useful – מוחקים את כל החוקים בהם הוא מופיע (דוגמא בדף).

2) אלג׳ למציאת המשתנים שניתנים להשגה מ- S:

$v(-1) = \emptyset$
$v(0) = \{S\}$
k = 0
while $v(k-1) \neq v(k)$ {
  k++
  $v(k) \leftarrow v(k-1) \cup \{A \in V, B \rightarrow \alpha A \beta, B \in v(k-1)\}$
}

כל משתנה שלא מופיע לבסוף ב-v, מבטלים את כל החוקים שבהם הוא מופיע (דוגמה בדף).

3) סילוק חוקי ε :

הרעיון : נמחק את חוקי ה- ε ונוסיף חוקים אחרים <u>כפיצוי</u> :

לדוגמא :

$A \rightarrow BC$
$S \rightarrow aA$
$C \rightarrow c \mid \varepsilon$
$B \rightarrow b \mid \varepsilon$

$\Rightarrow$

$B \rightarrow b$
$C \rightarrow c$
$A \rightarrow BC \mid \underline{B \mid C}$
$S \rightarrow aA \mid \underline{a}$

<u>הגדרה:</u> משתנה B יקרא אפיס אם $B \rightarrow^* \varepsilon$.

אלגוריתם למציאת המשתנים האפיסים:

$Null \leftarrow \{A \mid A \to \varepsilon \in P\}$

כל עוד Null גדלה מבצעים :

$Null \leftarrow Null \cup \{A \in V \mid A \to x_1 x_2 \ldots x_n \in P,\ A \notin Null,\ x_1, x_2, \ldots, x_n \in Null\}$

כעת :

1. מסלקים את כל כללי ε מהדקדוק.
2. עבור כל חוק $B \to \beta$ מוסיפים את החוקים
$\{B \to \beta` \mid \beta \neq \varepsilon$ וגם $\beta`$ מתקבלת מ- β ע"י מחיקת משתנים אפיסים בכל האפשרויות $\}$
3. אם S הוא משתנה אפיס, מוסיפים כלל $S \to \varepsilon$ (דוגמא בדף).

בצוע שלב 3, שכן נוצרו משתנים מיותרים (דוגמא בדף).

סילוק חוקי יחידה :

פעולה זו מקצרת את שרשרות הגזירה (מקטינה את גובה עץ הגזירה).

לדוגמא :

$$\begin{array}{lcl} A \to B & & A \to de \\ B \to C & \Rightarrow & B \to de \\ C \to de & & C \to de \end{array}$$

בהינתן $A \in V$ נגדיר :

$$Unity(A) = \{B \mid A \to^* B,\ B \in V,\ B \neq A\}$$

במילים, Unity(A) היא קבוצת כל המשתנים שניתן לגזור מהמשתנה A ע"י חוקי יחידה בלבד (כאשר אין חוקי ε).

עבור כל משתנה $A \in V$:

1. נסלק את חוקי היחידה מהצורה $A \to B$ $(B \in V)$
2. נוסיף חוקים $\{A \to \alpha \mid \alpha \notin V,\ C \in Unity(A),\ C \to \alpha$ קיים חוק $\}$
לא בהכרח קיים חוק $A \to C$ (כמו בדוגמא הנ"ל).
(דוגמא בדף).
3. בצוע שלב 5 שכן נוצרו משתנים מיותרים.