שיעור 5
התזה של צרץ טיורינג ודקדוקים כלליים

5.1 היחס בין הכרעה וקבלה

משפט 5.1 כל שפה כריעה היא גם קבילה

כל שפה כריעה היא גם קבילה.

הוכחה: המכונה טיורינג שמכריעה את L גם מקבלת אותה. ■

נשאל שאלה. האם כל שפה קבילה היא גם כריעה? זאת שאלה שכרגע אין לנו מספיק כלים לענות עליה. נחזור לשאלה הזו בפרק הבא. לבינתיים נוכיח טענה חלשה יותר.

משפט 5.2

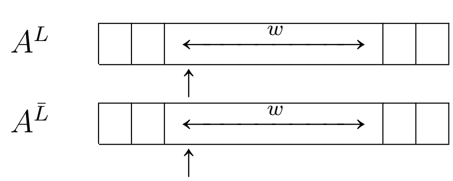
תהי L שפה.

אם גם L וגם $\bar{L}$ קבילות אזי L כריעה.

הוכחה: תהי A^L מ"ט שמקבלת את L, ותהי $A^{\bar{L}}$ מ"ט שמקבלת את $\bar{L}$.
נבנה מ"ט D^L שמכריעה את L.

<u>כיצד תעבוד המ"ט D^L המכריעה?</u>

- נריץ במקביל את A^L ואת $A^{\bar{L}}$.
- אם A^L מקבלת את המילה אז נעבור ל- acc.
- אם $A^{\bar{L}}$ מקבלת את המילה אז נעבור ל- rej.



■

- הסימולציה מתבצעת ע"י סימלוץ צעד צעד.
  * צעד במכונה A^L.
  * צעד במכונה $A^{\bar{L}}$.
- נמשיך בסימולציה המקבילית עד שאחת המכונות מגיעה למצב acc.

* אם A^L מקבלת ← acc.
* אם $A^{\bar{L}}$ מקבלת ← rej.

- לא יכול להיות מצב כי אף אחת מהמכונות לא מגיעה למצב acc כי כל מחרוזת $w \in L$ או $w \in \bar{L}$.

5.2 שקילות של מכונת טיורינג ותוכנית מחשב

- מכונת טיורינג היא מודל חישובי למחשב.
- מחשב = תוכנית מחשב.
- תוכנית משחב כתובה בשפת תכנות, למשל
  * ג'אווה
  * פייתון
  * C
  * SIMPLE
- המרכיבים של שפת תכנות הם
  * משתנים
  * פעולות
  * תנאים
  * זרימה

נוכיח כי מכונט טיורינג ותוכנית משחב שקולים חישובי.

5.3 SIMPLE

<u>משתנים</u>

- טבעיים:

```
i , j , k , . . . .
```

מקבלים כערך מספר טבעי.

- אתחול: הקלט נמצא בתאים הראשונים של `A[]`
  כל שאר המשתנים מאותחלים ל- 0.

<u>פעולות</u>

- השמה בקבוע:

```
i = 3 ,   B [ i ] = "#"
```

- השמה בין משתנים:

```
i = k ,     A [ k ] = B [ i ]
```

- פעולות חשבון:

```
x = y + z , x = y - z , x = y.z
```

<u>תנאים</u>

- `B[i]==A[j]` (מערכים).
- `x >= y` (משתנים טבעיים).

<u>זרימה</u>

- סדרה פקודות ממוספרות.
- goto : מותנה ולא מותנה.
- stop : עצירה עם ערך חזרה.

```
one = 1
zero = 0
B[zero] = "0"
i=0
j=i
if A[i] == B[zero] goto 9
i=j + one
goto 3
C[one] = A[j]
if C[one] == A[zero] goto 12
stop(0)
stop(1)
```

כעת נגדיר את מושגי הקבלה והדחייה של מילים בשפה SIMPLE,
ונגדיר את מושגי הכרעה והקבלה של שפות בשפה SIMPLE.

הגדרה 5.1 קבלה ודחייה של מחרוזת בשפה SIMPLE

עבור קלט w ותוכנית P בשפת SIMPLE. אומרים כי

- P **מקבלת** את w אם הריצה של P על w עוצרת עם ערך חזרה 1.
- P **דוחה** את w אם הריצה של P על w עוצרת עם ערך חזרה 0.

הגדרה 5.2 הכרעה וקבלה של מחרוזת בשפה SIMPLE

עבור שפה L ותוכנית P בשפת SIMPLE. אומרים כי

- P **מכריעה** את L אם היא מקבלת את המילים שב- L ודוחה את אלה שלא ב- L.
- P **מקבלת** את L אם היא מקבלת את כל ורק המילים ב- L.

משפט 5.3

המודלים של מכונת טיורינג ותוכניות SIMPLE שקולים.

הוכחה:

כיוון ראשון:

$$\forall M \, \exists P \text{ שקולה }.$$

במילים, לכל מ"ט M קיימת תוכנית P שקולה.
נבצע סימולציה של מ"ט M במשחב P .

בלי להכנס ,לפרטים די ברור שבשפה ,עילית כגון ג'אווה ,ניתן להגדיר מבני נתוניםעבור כל מרכיבי מכונת טיורינג:

- הסרט.
- המצבים.
- מיקום הראש.
- טבלת המעברים.

וברור שניתן לבצע סימולציה של פעילות המכונה.
ואם ניתן לעשות זאת בשפה עילית, ניתן לעשות זאת גם בשפת SIMPLE.

כיוון שני:

$$\forall P \, \exists M \text{ שקולה }.$$

במילים, לכל תוכנית P בשפה SIMPLE קיימת מ"ט שקולה.

אנחנו צריכים להראות כיצד ניתן לממש את הרכיבים השונים של תוכניות SIMPLE במ"ט.

הרכיבים הם:

- משתנים.
- פעולות.
- תנאים.
- זרימה.

משתנים

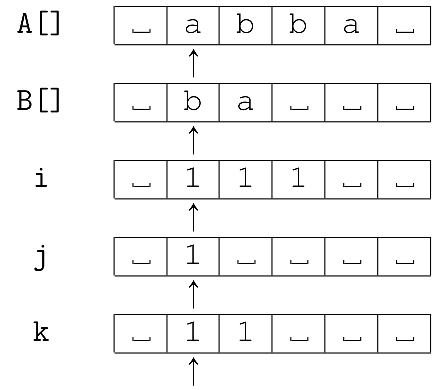
לכל משתנה יהיה סרט משלו.
המספר שהמשתנה יחזיק ייוצג בבסיס אונרי.
בהתחלה הסרט יהיה רק עם רווחים ,זה מייצג את המספר אפס בבסיס אונרי.

לכל מערך יהיה סרט משלו.
בכל תא בסרט המערך תהיה אות.
בהתחלה כל המערכים יהיו מאופסים למעט הסרט הראשון ,שיחזיק את הקלט.

למשל ההשמה הבאה של משתים בשפה SIMPLE:

```
A[1] = a, A[2] = b, A[3] = b , A[4] = a
B[1] = b, B[2] = a
i = 3
j = 1
k = 2
```

ניתן לממש במ"ט על ידי לכתוב על סרטים, שרט אחד לכל משתנה:

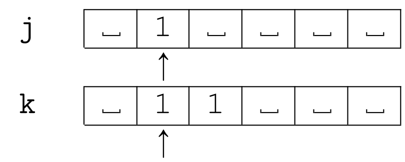


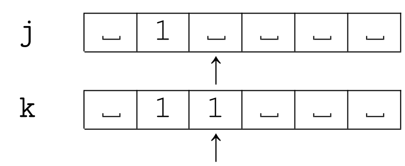
<u>פעולות</u>

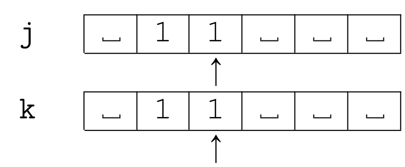
כעת נניח שנשים

j = k

אפשר לממש את ההשמה הזאת על ידי להעתיק את תוכן הסרט של המשתנה k לסרט של המשתנה j.

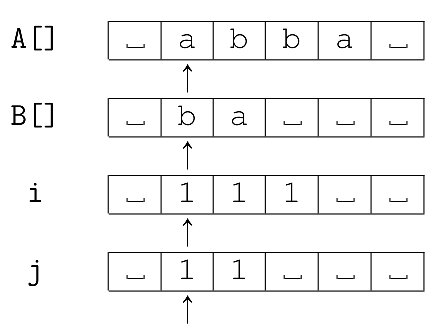




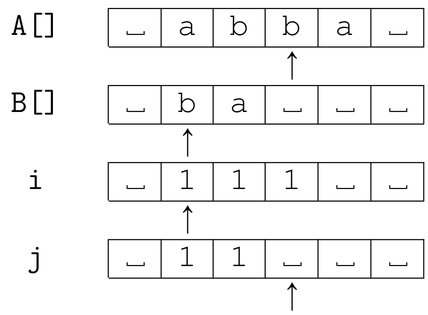


כעת נניח שנשים B[i]=A[j], ז"א B[3]=A[2].
נממש זה במ"ט ע"י להעתיק את תוכן משבצת 2 בסרט של A[] למשבצת 3 בסרט של B[].

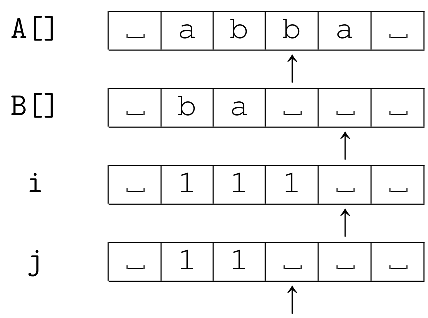
שלב 1)



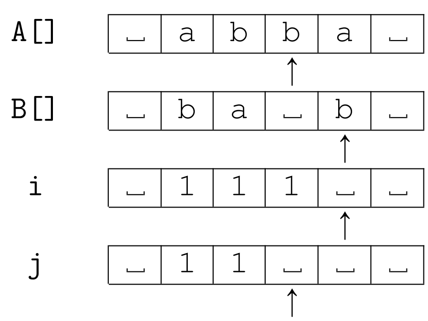
שלב 2)



שלב 3)



שלב 4)



נניח עכשיו שאנחנו רוצים לשים

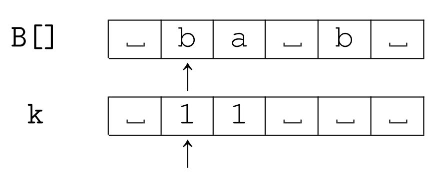
```
B[k] <- "a"
```

ז"א

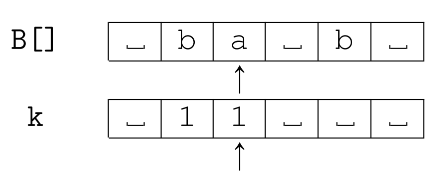
```
B[3] <- "a"
```

נממש זה במ"ט ע"י על ידי הפעולות הבאות עם הסרט של B[] והסרט של k.

שלב 1)



שלב 2)



שלב 3)

B[]	␣	b	a	␣	b	␣
				↑		
k	␣	1	1	␣	␣	␣
				↑		

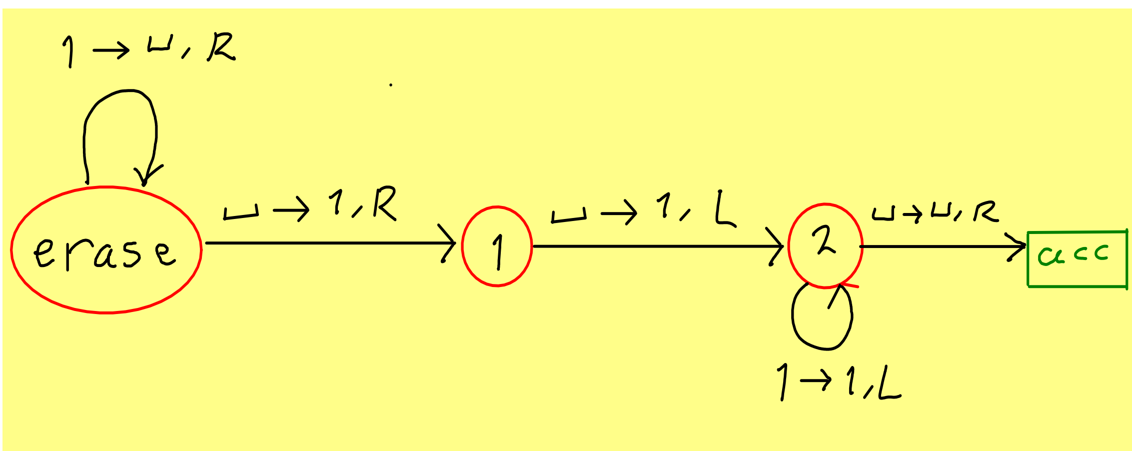
שלב 4)

B[]	␣	b	a	a	b	␣
				↑		
k	␣	1	1	␣	␣	␣
				↑		

כעת נניח שאנחנו רוצים לשים

j = 2

אז נממש זה במ"ט עם הפעולות הבאות:



שלב 1)

j	␣	1	␣	␣	␣	␣
		↑				

שלב 2)

j	␣	␣	␣	␣	␣	␣
			↑			

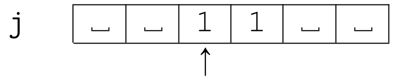
שלב 3)

j	␣	␣	1	␣	␣	␣
			↑			

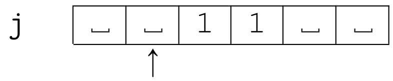
שלב 4)

j	␣	␣	1	1	␣	␣
				↑		

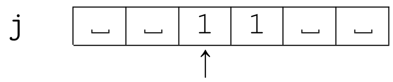
שלב 5)



שלב 6)



שלב 7)



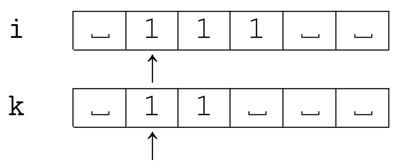
<u>תנאים</u>

נניח שאנחנו רוצים לממש את התנאי

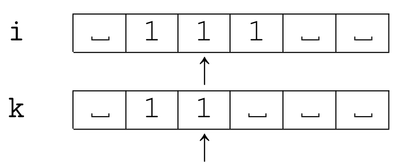
i >= k

ניתן לבדוק את התנאי במ"ט על ידי הפעולות הבאות:

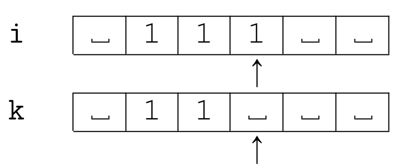
שלב 1)



שלב 2)



שלב 3)



■

5.4 דקדוקים כלליים

הגדרה 5.3 דקדוקים חסרי קשר

דקדוק חסר קשר הוא קבוצה
$$(V, \Sigma, R, S)$$
כאשר

- V קבוצה סופית של **משתנים** שמורכב מאותיות גדולות של אלפיבית.
- Σ קבוצה סופית של **טרמינלים** שמורכב מאותיות קטנות וסימנים של אלפיבית.
- R קבוצה של כללים. כל כלל הוא מצורה
$$\gamma \to u$$
כאשר $\gamma \in V$ משתנה בודד בצד שמאל ו- $u \in (V \cup \Sigma)^*$ מחרוזת של משתנים וטרמינילים בצד ימין
- $S \in V$ המשתנה ההתחלתי.

דוגמה 5.1

נתון הדקדוק חסר קשר:
$$G_1 = (\{A, B\}, \{0, 1, \#\}, R, A)$$
הקבוצת משתנים היא $V = \{A, B\}$, הקבוצת טרמינלים היא $V = \{0, 1, \#\}$, המשתנה ההתחלתי הוא $S = A$ והכללים של הדקדוק הם
$$R = \begin{cases} A \to & 0A1 \\ A \to & B \\ B \to & \# \ . \end{cases}$$

הגדרה 5.4 יצירה של מילה על ידי דקדוק חסר קשר

1) כתבו את המשתנה ההתחלתי S.

2) מצאו משתנה וכלל אשר מתחיל אם משתנה זה, והחליפו אותו עם המחרוזת בצד ימין של הכלל.

3) חזרו על שלבים 1 ו- 2 עד שלא נשאר אף משתנים של V.

דוגמה 5.2

הדקדוק G_1 יוצר את המחרוזת $000\#111$:
$$A \xrightarrow{A\to 0A1} 0A1 \xrightarrow{A\to 0A1} 00A11 \xrightarrow{A\to 0A1} 000A111 \xrightarrow{A\to B} 00B11 \xrightarrow{B\to \#} 000\#111$$

דוגמה 5.3

נתון את הדקדוק
$$G_2 = (\{S, T, F\}, \{(,), +, \times, a\}, R, S)$$
כאשר הכללים הם

$$R = \begin{cases} S \to & S+T \\ S \to & T \\ T \to & T\times F \\ T \to & F \\ F \to & (S) \\ F \to & a\ . \end{cases}$$

G_2 יוצר את המילה: $a+a$:

$$S \xrightarrow{S\to S+T} S+T \xrightarrow{SA\to T} T+T \xrightarrow{T\to F} F+F \xrightarrow{F\to a} a+a$$

בדקדוק כללי, גם בצד ימין וגם בצד שמאל יכולה להופיעמחרוזת של משתנים וטרמינליים. פורמלי:

הגדרה 5.5 דקדוקים כלליים

דקדוק חסר קשר הוא קבוצה

$$(V,\Sigma,R,S)$$

כאשר

- V קבוצה סופית של **משתנים** שמורכב מאותיות גדולות של אלפיבית.
- Σ קבוצה סופית של **טרמינלים** שמורכב מאותיות קטנות וסימנים של אלפיבית.
- R קבוצה של כללים. כל כלל הוא מצורה

$$\gamma \to u$$

כאשר $\gamma \in (V\cup\Sigma)^+$, $u\in(V\cup\Sigma)^*$. מחרוזת של משתנים וטרמינילים בצד ימין

- $S\in V$ המשתנה ההתחלתי.

דוגמה 5.4

נתון את הדקדוק

$$G = (\{\texttt{S},\texttt{[},\texttt{]}\},\{\texttt{a}\},R,\texttt{S})$$

שבו הקבוצת משתנים היא $V=\{\texttt{S},\texttt{[},\texttt{]}\}$, הקבוצת טרמינליים היא $\Sigma=\{\texttt{a}\}$ והכללים הם

$$R = \begin{cases} \texttt{S} \to & \texttt{[S]} \\ \texttt{S} \to & \texttt{a} \\ \texttt{[a} \to & \texttt{aa[} \\ \texttt{[]} \to & \varepsilon\ . \end{cases}$$

G יוצר את המילה: $\texttt{aaaa}$:

$$\texttt{S} \xrightarrow{\texttt{S}\to\texttt{[S]}} \texttt{[S]} \xrightarrow{\texttt{S}\to\texttt{[S]}} \texttt{[[S]]} \xrightarrow{\texttt{S}\to\texttt{a}} \texttt{[[a]]} \xrightarrow{\texttt{[a}\to\ \texttt{aa[}} \texttt{[aa[]]}$$
$$\xrightarrow{\texttt{[]}\ \to\varepsilon} \texttt{[aa]} \xrightarrow{\texttt{[a}\to\ \texttt{aa[}} \texttt{aa[a]} \xrightarrow{\texttt{[a}\to\ \texttt{aa[}} \texttt{aa\ aa[]} \xrightarrow{\texttt{[]}\ \to\varepsilon} \texttt{aaaa}$$

דוגמה 5.5

נתון את הדקדוק

$$G = (\{\mathtt{S}, \mathtt{[}, \mathtt{]}\}, \{\mathtt{a}\}, R, \mathtt{S})$$

שבו הקבוצת משתנים היא $V = \{\mathtt{S}, \mathtt{[}, \mathtt{]}\}$, הקבוצת טרמינליים היא $\Sigma = \{\mathtt{a}\}$ והכללים הם

$$R = \begin{cases} \mathtt{S} \to & \mathtt{[S]} \\ \mathtt{S} \to & \mathtt{a} \\ \mathtt{[a} \to & \mathtt{aa[} \\ \mathtt{[]} \to & \varepsilon\ . \end{cases}$$

מהן המילים שניתן לצור בעזרת הדקדוק הזה,
או במילים אחרות :מהי השפה של הדקדוק?

פתרון:

תשובה:

$$L(G) = \{\mathtt{a}^n \mid n = 2^k\ ,\ k \geqslant 1\}\ .$$

הסבר:

$$\begin{array}{lllllll}
\mathtt{S} & \xrightarrow{\mathtt{S}\to\mathtt{[S]}} & \mathtt{[S]} & \xrightarrow{\mathtt{S}\to\mathtt{[S]}} & \cdots & \xrightarrow{\mathtt{S}\to\mathtt{[S]}} & \mathtt{[}^k\mathtt{S]}^k \\
 & \xrightarrow{\mathtt{S}\to\mathtt{a}} & \mathtt{[}^k\mathtt{a]}^k & & & & \\
 & \xrightarrow{\mathtt{[a}\to\ \mathtt{aa[}} & \mathtt{[}^{k-1}\mathtt{aa[]}^k & \xrightarrow{\mathtt{[]}\to\varepsilon} & \mathtt{[}^{k-1}\mathtt{aa]}^{k-1} & & \\
 & \xrightarrow{\mathtt{[a}\to\ \mathtt{aa[}} & \mathtt{[}^{k-2}\mathtt{aa[a]}^{k-1} & \xrightarrow{\mathtt{[a}\to\ \mathtt{aa[}} & \mathtt{[}^{k-2}\mathtt{aaaa[]}^{k-1} & \xrightarrow{\mathtt{[]}\to\varepsilon} & \mathtt{[}^{k-2}\mathtt{aaaa]}^{k-2} \\
 & & & & \cdots & \to & \mathtt{a}^k
\end{array}$$

דוגמה 5.6

בנו דקדוק כללי אשר יוצר את הפשה

$$L = \{w \in \{a,b\}^* \mid \#a_w = \#b_w\}\ .$$

פתרון:

$$G = (\{\mathtt{S}\}, \{\mathtt{a}, \mathtt{b}\}, R, \{\mathtt{S}\})$$

$$S \to \mathtt{abS}\ , \tag{1}$$

$$\mathtt{ab} \to \mathtt{ba}\ , \tag{2}$$

$$\mathtt{ba} \to \mathtt{ab}\ , \tag{3}$$

$$\mathtt{S} \to \varepsilon\ . \tag{4}$$

$$\mathtt{S} \xrightarrow{1} \mathtt{abS} \xrightarrow{1} \mathtt{ababS} \xrightarrow{2} \mathtt{baabS} \xrightarrow{4} \mathtt{baab}$$

שימו לב: בדקדוק כללי אנו מאפשרים גם כללייצירה בהם בצד שמאל יש רק טרמינלים.
לכן ,יתכן גם שנמשיך ונפתח מחרוזתשכולה טרמינליים. למשל

$$\texttt{S} \xrightarrow{1} \texttt{abS} \xrightarrow{1} \texttt{ababS} \xrightarrow{4} \texttt{abab} \xrightarrow{2} \texttt{baab}$$

נשאל שאלה כללית:

- אלו שפות ניתן לצור בעזרת דקדוק כללי?
- האם יש שפות שלא ניתן לצור בעזרת דקדוק כללי?
- האם יש מודל חישובי שמקבל שפות שנוצרות ע"י דקדוקים כלליים?

דוגמה 5.7

בנו דקדוק כללי שיוצר את השפה

$$w = \{w \in \{\mathrm{a}, \mathrm{b}, \mathrm{c}\}^* \mid w = \mathrm{a}^n \mathrm{b}^n \mathrm{c}^n\}$$

פתרון:
נראה דקדוק כללי עבור שפה זו.
שפה זו אינה חסרת הקשר.
לכן ,לא ניתן לבנות עבורה דקדוק חסר הקשר.
אנו נבנה לה דקדוק כלל.
נגזור את האותיות a,b,c יחד.
נעשה זאת בצורה כזו שכדי לסיים את תהליך הגזירה יש לסדר את האותיות בסדר הרצוי:
תחילה a,
אחר כך b,
ובסוף c.

$$\texttt{S} \to \texttt{S']} \tag{1}$$
$$\texttt{S'} \to \texttt{aS'bC} \mid \varepsilon \tag{2}$$
$$\texttt{Cb} \to \texttt{bC} \tag{3}$$
$$\texttt{C]} \to \texttt{]c} \tag{4}$$
$$\texttt{]} \to \varepsilon \tag{5}$$

$$\begin{aligned}
\texttt{S} &\xrightarrow{1} \texttt{S']} \xrightarrow{2} \texttt{aS'bC]} \xrightarrow{2} \texttt{aaS'bCbC]} \xrightarrow{2} \texttt{aaaS'bCbCbC]} \\
&\xrightarrow{3} \texttt{aaaS'bbCCbC]} \xrightarrow{3} \texttt{aaaS'bbCbCC]} \xrightarrow{3} \texttt{aaaS'bbbCCC]} \\
&\xrightarrow{4} \texttt{aaaS'bbbCC]c} \xrightarrow{4} \texttt{aaaS'bbbC]cc} \xrightarrow{4} \texttt{aaaS'bbb]ccc} \\
&\xrightarrow{5} \texttt{aaaS'bbbccc} \xrightarrow{1} \texttt{aaabbbccc}
\end{aligned}$$

דוגמה 5.8

בנו דקדוק כללי אשר יוצר את שפת המילים

$$L = \{\ \mathrm{uu} \mid \mathrm{u} \in \{\mathrm{a},\mathrm{b}\}^*\}$$

פתרון:

דוגמא זאת תמחיש ביצד דקדוק כללי יכול "לפעול בדומה" למכונת טיורינג.
בדקדוק נשתמש במשתנים וכללי גזירה שיאפשרו מעין תנועה על גבי המחרוזת הנגזרת, בדומה לתנועת הראש של מכונת טיורינג על גבי הסרט.

1	כלל גזירה יחיד מהמשתנה ההתחלתי. המשתנה H ידמה את הראש של המ"ט ש"יזוז" מצד לצד על המחרוות הנגזרות. הסוגר המרובע] מסמן את הקצה השמאלי של המילה השמאלית. הסוגר המסולסל } מסמן את הקצה השמאלי של המילה הימינית.	$S \to [H\{$
2	כלל זה מאפשר הוספת אות a לקצה השמאלי של המילה השמאלית. המשתנה H מוחלף במשתנה H_a כדי "לזכור" שיש עכשיו להוסיף a גם במחרוזת הימינית. (בדומה לזיכרונות של מ"ט).	$[H \to [aH_a$
3	כלל זה מאפשר לראש "לזוז" ימינה.	$H_a a \to aH_a$
4	כאשר המשתנה H_a "יגיע" לסוגר המסולסל, הוא יגזור אות a נוספת מימין לסוגר, שהוא הקצה השמאלי של המחרוזת הימינית. כך אפשרנו להוסיף שתי אותיות a: אחת מימין לסוגר] ואחת תואם ימין לסוגר }. כלומר אות a בקצה השמאלי של כל אחת המחרוזות.	$H_a\{ \to H\{a$
5	כעת צריך לאפשר למשתנה H לחזור שמאלה על גבי האותיות שבין הסוגרים, עד הסוגר].	$aH \to Ha$

ברגע "שהראש" H חזר לתחילת המחרוזת ועומד ליד הסוגר] עוברים על השלבים 2-5 שוב. בסבב הבא נחק בחשבון גם יצירה של שתי אותיות b.

2'	כלל זה מאפשר הוספת אות b לקצה השמאלי של המילה השמאלית. המשתנה H מוחלף במשתנה H_b כדי "לזכור" שיש עכשיו להוסיף b גם במחרוזת הימינית.	$[H \to [bH_b$
3'	כללים האלה מאפשרים לראש "לזוז" ימינה.	$H_a a \to aH_a$ $H_a b \to bH_a$ $H_b a \to aH_b$ $H_b b \to bH_b$
4'	כאשר המשתנה H_b "יגיע" לסוגר המסולסל, הוא יגזור אות b נוספת מימין לסוגר, שהוא הקצה השמאלי של המחרוזת הימינית. כך אפשרנו להוסיף שתי אותיות b: אחת מימין לסוגר] ואחת תואם ימין לסוגר }. כלומר אות b בקצה השמאלי של כל אחת המחרוזות.	$H_b\{ \to H\{b$
5'	כעת צריך לאפשר למשתנה H לחזור שמאלה על גבי האותיות שבין הסוגרים, עד הסוגר].	$bH \to Hb$

בכדי לסיים את הגזירה יש להפטר ממשתני העזר על ידי הכללים הבאים:

6	הכללים האלה מאפשרים להעלים את המשתנים `H, [, {`	$H \to \varepsilon$ $[\to \varepsilon$ $\{ \to \varepsilon$

למשל:

$$S \xrightarrow{1} [H\{ \xrightarrow{2} [aH_a\{ \xrightarrow{4} [\ aH\{a \xrightarrow{5} [\ Ha\{a$$
$$\xrightarrow{2} [aH_a a\{a \xrightarrow{3} [aaH_a\{a \xrightarrow{4} [aa\ H\{aa \xrightarrow{5} [Haa\ \{aa$$
$$\xrightarrow{2} [bH_b aa\{aa \xrightarrow{3} [baaH_b\{aa \xrightarrow{4} [baa\ H\{baa \xrightarrow{5} [Hbaa\ \{baa$$
$$\xrightarrow{6} baabaa$$

5.5 דקדוקים כלליים ומכונת טיורינג

משפט 5.4 קדוקים כלליים ומכונת טיורינג

תהי L שפה. L קבילה אם ורק אם קיים דקדוק כללי G כך ש- $L(G) = L$.

הוכחה: כיוון ראשון.

נוכיח שאם קיים דקדוק כללי G אז $L(G)$ קבילה.

נניח שקיים דקדוק כללי G. נוכיח כי $L(G)$ קבילה על ידי להוכיח שקיימת תוכנית מחשב P שמקבלת $L(G)$.

נתון דקדוק כללי G. נבנה תוכנית מחשב שמקבלת את $L(G)$.
יהי הקלט $w \in L(G)$, כלומר w מילה בשפה $L(G)$.

(1 u=S.

(2 repeat:

- פצל באופן לא דטרמיניסטי את u ל- xyz.
- בחר באופן לא דטרמיניסטי גזירה $\mathtt{v} \to \mathtt{t}$ של G.
- אם $\mathtt{t} \neq \mathtt{y}$ דחה.
- u=xvz
- אם w==u קבל.

כיוון שני.

נוכיח שאם $L(G)$ קבילה אז קיים דקדוק כללי G.

ז"א, נניח שקיימת מ"ט M שמקבלת את השפה L. נוכיח שקיים דקדוק כללי G כך ש- $L(G) = L(M)$, כלומר השפה המתקבלת על ידי M היא השפה של דקדוק כללי G.

נתונה מ"ט M בעלת הטבלת המעברים להלן. נבנה דקדוק כללי G שממשש אותם צעדים.

מצב	סימן	מצב חדש	כתיבה	תזוזה
q_0	a	q_0	a	R
q_0	b	q_1	a	R
q_0	␣	acc	␣	L
q_1	a	q_0	b	L
q_1	b,␣	q_1	b	L

לפי הטבלת המעברים קיים הצעד

$$\mathtt{q}\ q_0\ \mathtt{b}\ \ \mathtt{a}\ \ \mathtt{b} \vdash_M \ \ \mathtt{aa}q_1\ \mathtt{ab}$$

נניח שבדקדוק כללי G קיים אותו הצעד

$$\mathtt{q}\ q_0\ \mathtt{bab} \xrightarrow{G} \ \ \mathtt{aa}q_1\ \mathtt{ab}$$

ניתן לממש צעד זה על ידי הכלל

$$q_0\ \mathtt{b} \to \mathtt{a}\ q_1$$

באופן כללי,

- עבור כל פונקצית המעברים של M שגוררת תזוזה ימינה מצורה

  $$\delta(q,\sigma) = (p,\pi,R)$$

  נממש מעבר זה על ידי כלל של הדקדוק G מצורה

  $$q\sigma \to \pi p\ .$$

- עבור כל פונקצית המעברים של M שגוררת תזוזה שמאלה מצורה

  $$\delta(q,\sigma) = (p,\pi,L)$$

  אז לכל $\tau\in\Gamma$ ב- G נממש מעבר זה על ידי הכלל

  $$\tau q\sigma \to p\tau\pi\ .$$

■

5.6 ההירככיה של חומסקי

משפחת שפות	דקדוק	מודל חישובי
קבילות	כללי	מכונת טיורינג
חסרות הקשר	חסר הקשר	אוטומט מחסנית
רגולריות	רגולרי	אוטומט סופי

- הירככיה של חומסקי קושרת לנו בין משפחותשל שפות דקדוקים ומודלים חישוביים.

- בתחתית ההיררכיה נמצאות השפות הרגולריות שנוצרות על ידי דקדוקים רולריים ומתקבלות על ידי אוטומטים סופיים.
- מעליהן נמצאות השפות חסרות ההקשר שנוצרות על ידי דקדוקים חסרי הקשר ומתקבלות על ידי אוטומטי מחסנית.
- ומעליהן נמצאות השפות הקבילות שנוצרותעל ידי דקדוקים כלליים ומתקבלותעל ידי מכונות טיורינג.
- כל רמה בהיררכיה מכילה ממש את הרמה שמתחתיה.
  * כל שפה רגולרית היא גם חסרתהקשר ,אבל יש שפות חסרות הקשרשאינן רגולריות.
  * כל שפה חסרת הקשר היא קבילה, אבל יש שפות קבילות שאינן חסרות הקשר.

5.7 כל שפה חסרת הקשר הינה כריעה

לפי ההיררכיה של חומסקי אנחנו יודעים לקבוע שכל שפה חסרת הקשר היא קבילה.

האם כל שפה חסרת הקשר הינה כריעה?

משפט 5.5

יהי $G=(V,\Sigma,S,R)$ דקדוק חסר הקשר ו- $w\in L(G)$. אזי קיים עץ גזירה של w שעומקו לכל היותר $(|V|+1)(|w|+1)$.

הוכחה: יהי T עץ הגזירה הקטן ביותר (מבחינת מספר קדקודים) של w.

בשלילה נניח שב- T יש מסלול מהשורש לעלה שמכיל לפחות $(|V|+1)(|w|+1)$ קודקודים פנימיים.

נסמן מסלול זה ב-

$$p=(u_1,u_2,\ldots,u_m)\ .$$

עבור קודקוד u_i במסלול נסמן ב- $s(u_i)$ את תת-המחרוזת של w שנוצרת מ- u_i.

מתקיים ש- $s(u_{i+1})$ היא תת-מחרוזת של $s(u_i)$. אומרים שקדקוד u_i הוא משמעותי אם $s(u_i)$ מכיל ממש את $s(u_{i+1})$.

כל קודקוד משמעותי מוסיף לפחות אות אחת ל- w.

לכן, ישנם לכל היותר $|w|$ קודקודים משמעותיים.

לכן, ברצף הקודקודים הפנימיים $(u_1,u_2,\ldots,u_m)$ שאורכן לפחות $(|V|+1)(|w|+1)$, בהכרח ישנו תת רצף $(u_i,u_{i+1},\ldots,u_{i+|V|})$ באורך $|V|+1$, שבו כל הקודדוקים לא משמעותיים.

ברצף זה בהכרח ישנם שני קודקודים, נאמר u_j,u_k, $j<k$ שמסומנים עם אותו משתנה.

לכן בעץ הגזירה, ניתן להחליף את הקודקוד u_j יחד עם כל תת העץ שמתחתיו- בקודקוד u_k, יחד עם כל תת העץ שמתחתיו.

כיוון שכל הקודקודים שבין u_j ל- u_k (כולל) הם לא משמעותיים, **החלפה זו לא משנה את המחרוזת הנוצרת.**

כלומר, העץ החדש גם הוא עץ הגזירה עבור w.

בסתירה להנחת המינימליות של העץ.

■

משפט 5.6

כל שפה חסרת הקשר היא כריעה.

הוכחה: בהינתן דקדוק חסר הקשר $G = (V, \Sigma, S, R)$, התוכנית הלא דטרמיניס הבאה מכריעה את $L(G)$.

קלט: מחרוזת w.
פלט: כן או לא.

1) נחש עץ גזירה של הדקדוק G בעומק לכל היותר $(|V| + 1)(|w| + 1)$.

2) בדוק האם העץ יוצר את המחרוזת w. אם כן, החזר "כן" איתר החזר "לא".

שני שלבי התוכנית בהכרח מסתיימים. לכן, זו תוכנית להכרעה. ישנו חישוב שמחזיר "כן" אם ורק אם $w \in L(G)$ לכן זו תוכנית שמכריעה את $L(G)$.

■