

# PARSING USING PC-PATR – דקדוק חסר הקשר

## PC-PATR

ניתן להוריד באופן חופשי מאת:

<http://www.sil.org/pcpatr>

מנתח תחבירי עבור דקדוקים חסרי הקשר ודקדוקי האחדה.

התוכנה קולטת דקדוק חסר הקשר בשני קבצים – קובץ הלקסיקון וקובץ הדקדוק, ולאחר מכן מבצעת ניתוח למחרוזות שונות לבחירתנו כולל מתן כל עצי הגזירה האפשריים.

[קובץ הלקסיקון](#):

מכיל את כל הטרמינלים (כלומר, את כל המילים ב- $\Sigma$ ).

יש להכניס כל מילה בפורמט הבא:

<code>\w word</code>	←	כאן יש לכתוב את הטרמינל
<code>\c something</code>	←	(קטגוריית המילה - כאן יש לכתוב דבר כלשהוא (כדאי את הנון-טרמינל)
<code>\g</code>	}	את השדות הללו (word gloss), (additional features)
<code>\f</code>		

○ בסוף שורה אין לשים סימן פיסוק.

○ יש לכתוב לקובץ את המילים, כל מילה בצורה הני"ל. מומלץ (אך אין הכרח) להפריד בין מילה למילה ע"י שורה רווח.

**דוגמא:**

```
\w fox
\c N
\g canine
\f <number> = singular
```

```
\w foxes
\c N
\g canine+PL
\f <number> = plural
```

[קובץ הדקדוק](#):

מכיל את כל החוקים בשפה.

כל חוק נכתב בשורה נפרדת ללא סימן פיסוק בסופו.

כל חוק ייכתב בצורה:

Rule NonTerminal -> RuleBody

המילה Rule יכולה להיכתב כ- rule או RULE.

לטרמינלים ונון-טרמינלים בחוקים יש את התכונות הבאות:

1. Case sensitive – רגישות להבדל בין אותיות קטנות וגדולות. לדוגמא: NOUN ו- Noun יהיו נון-טרמינלים שונים.

2. באות X ניתן להשתמש כתחליף לכל טרמינל או נון-טרמינל.  
לדוגמא: החוק הבא בדקדוק יאמר שאפשר להחליף כל קטגוריה בדקדוק בשני עותקים של אותה קטגוריה מופרדים במלת חיבור (CJ for conjunction)

```
Rule X -> X_1 CJ X_2  
<X cat> = <X_1 cat>  
<X cat> = <X_2 cat>  
<X arg1> = <X_1 arg1>  
<X arg1> = <X_2 arg1>
```

3. הסימן \_ (underscore) ואחריו אינדקס משמשים להבדיל בין מופעים של אותו סימן כמה פעמים בחוק.  
לדוגמא:

The rule NP -> P NP should be written as:

Rule NP -> P NP\_1

4. הסימנים {} [] <> := / שמורים. אין להשתמש בהם בטרמינלים או נון-טרמינלים.  
בסימן \_ ניתן להשתמש רק על פי הנאמר בסעיף 3.  
5. הסימן השמאלי בחוק הראשון בקובץ הדקדוק יהיה סימן ההתחלה בדקדוק (S).

לטרמינלים ונון-טרמינלים בצד ימין של חוקים יש את האפשרויות הבאות:

1. סוגריים עגולים מסביב לטרמינלים או נון-טרמינלים הופכים אותם לאופציונאליים.  
2. קו נטוי (forward slash) משמש ל-OR, ז"א להפרדה בין ביטויים שהם הצד הימני של אותו החוק. לדוגמא:

Rule NP-> (ADJ) N / DET (ADJ) N

3. סוגריים מסולסלים משמשים לקיבוץ סימנים. לדוגמא:

Rule S -> NP {TVP / IV}

=> S-> NP TVP , S-> NP IV

קליטת קבצים וניתוח

ראשית יש לתת לתוכנה לקלוט את הקבצים.

שימו לב שכל סמל שיופיע בקובץ הדקדוק אך לא בקובץ הלקסיקון יזוהה כנון-טרמינל.

קליטת קובץ הלקסיקון מתבצעת ע"י הפקודה:  
load lexicon filename או:  
קליטת קובץ הדקדוק מתבצעת ע"י הפקודה:  
load grammar filename או:

כעת אנו יכולים לבצע ניתוח על משפטים.

ע"י הפקודה `<parse <string` ניתן לבצע ניתוח למשפט בודד.

הפלט יהיה תוצאת שייכות או אי שייכות לשפה, ובמידה והמשפט בשפה יודפסו לו כל עצי הניתוח האפשריים.

כמו כן ניתן לתת קובץ המכיל מספר משפטים לניתוח ע"י הפקודה

`[file parse input-file [output-file`

במידה וניתן קובץ פלט יודפסו אליו הניתוחים, במידה ולא, הם יופיעו על המסך.

סיום

כדי לצאת מהתוכנה יש להשתמש בפקודה `.exit`.

## דוגמאות:

דוגמא 1:

עבור השפה  $L = \{a^n b^n \mid n \geq 0\}$  יש לבנות את הדקדוק הבא:

$$S \rightarrow aSb$$

$$S \rightarrow \epsilon$$

קובץ הלקסיקון:

`\w a`

`\c a`

`\g`

`\f`

`\w b`

`\c b`

`\g`

`\f`

קובץ הדקדוק:

Rule S -> (a S\_1 b)

דוגמא 2:

עבור השפה  $L = \{a^n b^m c^{n+m} \mid n, m > 0\}$  יש לבנות את הדקדוק הבא:

$$S \rightarrow aSc \mid aAc$$

$$A \rightarrow bAc \mid bc$$

קובץ הלקסיקון:

\w a  
\c a  
\g  
\f  
\w b  
\c b  
\g  
\f  
\w c  
\c c  
\g  
\f

קובץ הדקדוק:

Rule S -> a S\_1 c / a A c  
Rule A -> b A\_1 c / b c

דוגמא 3:

קובץ לקסיקון:

\w the  
\c the  
\g  
\f  
\w cat  
\c cat  
\g  
\f  
\w hat  
\c hat  
\g  
\f  
\w in  
\c in  
\g  
\f

קובץ דקדוק:

Rule NP -> D N / NP\_1 PP  
Rule PP -> P NP  
Rule D -> the  
Rule N -> cat / hat  
Rule P -> in