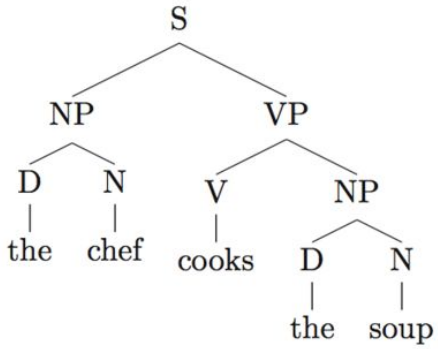
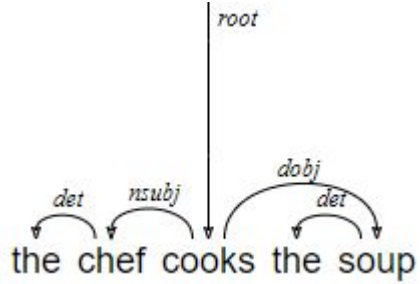
עיבוד שפה טבעית ש10:
ניתוח מרכיבים ותלויות
(SLP 17-18, E 10-11)

תזכורת - שבוע הבא בזום

ניתוח תחבירי Syntactic Parsing





- כמה עצי מרכיבים אפשריים יש למשפט בן n מילים?
  - אינסוף
- כמה עצי CNF אפשריים יש למשפט בן n מילים?
  - עצרתי ב-2n - מספר קטלאן של n-1 (ר' אייזנסטיין)
- כמה עצי תלויות אפשריים יש למשפט בן n מילים?
  - עצרתי "רגיל" n^(n-2)
- כמה עצי תלויות היטליים אפשריים יש למשפט בן n מילים?
  - חזקתי O(2^n)

ניתוח מרכיבי של CNF

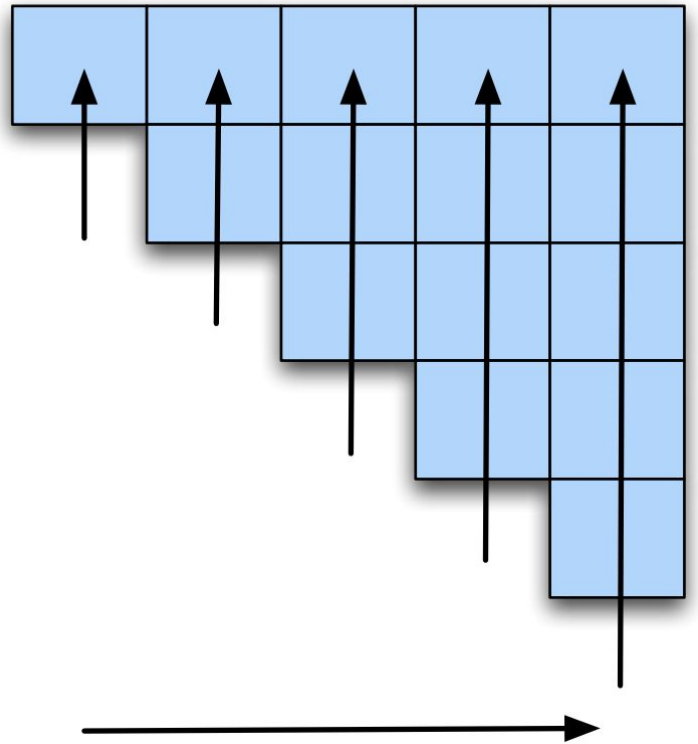
- בהינתן סט חוקים ומשפט, יש סיכוי גדול שאפשר לנתח אותו בכמה דרכים שונות.
- כמו עם תיוג, נוכל לפרק את הניתוח המלא לחלקים ולתת ציון (אולי הסתברותי) לכל ניתוח חלקי
- כלל הציון (scoring function) צריך להיות **קומפוזיציונלי** - ציון למרכיב מסוים מסתמך על הציון של חלקיו - ואז נוכל להרכיב אותו מלמטה מעלה.

אלגוריתם CKY (לא למבחן)

- תכנות דינמי, אנו נפגשים שוב.
- מייצגים את המשפט ע״י צמתיו האפשריים במטריצה משולשית עליונה שבה כל תא מייצג רצף (span) של טקסט
- בכל תא נרשום את כל הניתוחים האפשריים לרצף שאותו הוא מייצג

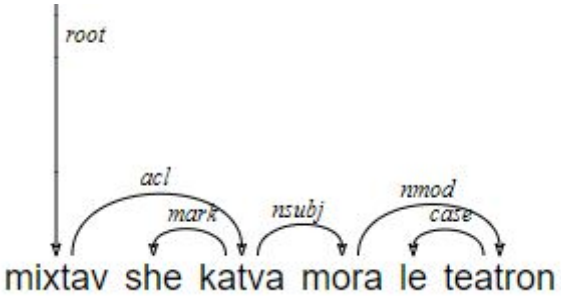
Book	*the*	*flight*	*through*	*Houston*
S, VP, Verb, Nominal, Noun [0,1]	[0,2]	S,VP,X2 [0,3]	[0,4]	S,VP,X2 [0,5]
	Det [1,2]	NP [1,3]	[1,4]	NP [1,5]
		Nominal, Noun [2,3]	[2,4]	Nominal [2,5]
			Prep [3,4]	PP [3,5]
				NP, Proper-Noun [4,5]

אלגוריתם CKY (לא למבחן)

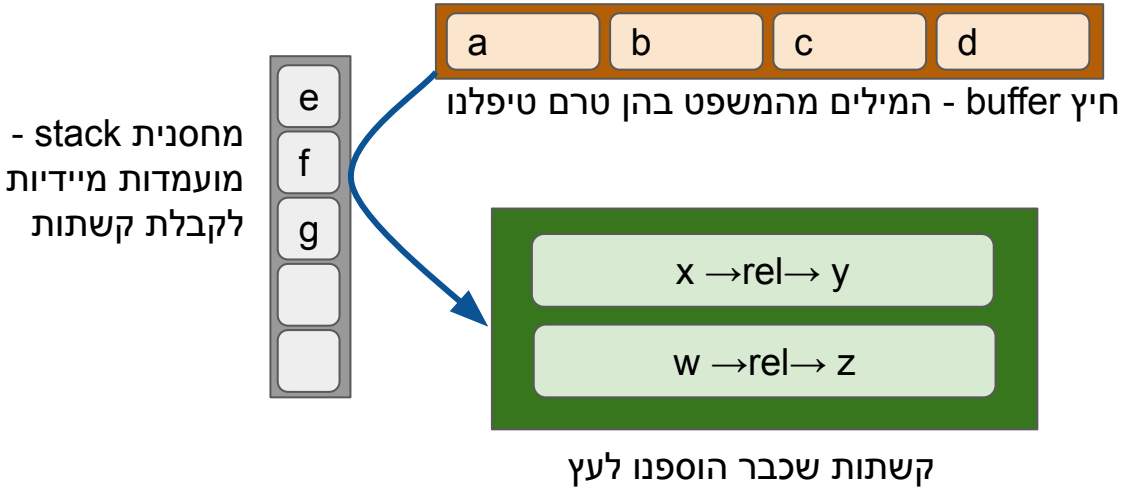


- תכנות דינמי, אנו נפגשים שוב.
- מייצגים את המשפט ע״י צמתיו האפשריים במטריצה משולשית עליונה שבה כל תא מייצג רצף (span) של טקסט
- בכל תא נרשום את כל הניתוחים האפשריים לרצף שאותו הוא מייצג
- נבנה את העץ במעבר על עמודות המטריצה משמאל לימין, כל עמודה מלמטה למעלה

ניתוח תלויות במעברים (Transition-based dependency parsing)



- בניית העץ (כמעט) "משמאל לימין", קשת אחרי קשת
- נחזיק שלושה מבנים שבאמצעותם ייבנה העץ:



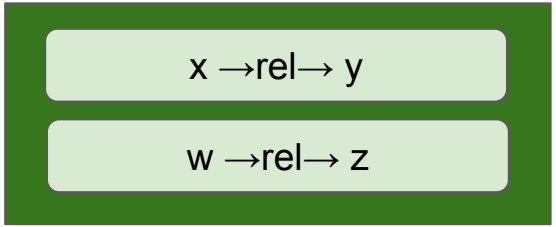
- מצב התחלתי?
- מצב סופי?
- סיבוכיות?

מעברים

- הזח SHIFT - העברת מילה מהחיץ למחסנית
- קשת שמאלית LEFT-ARC - הוסף קשת מהמילה העליונה במחסנית לשניה והסר את השניה
- קשת ימנית RIGHT-ARC - הוסף קשת מהמילה השניה במחסנית לעליונה והסר את העליונה

חיץ buffer - המילים מהמשפט בהן טרם טיפלנו

מחסנית stack - מועמדות מיידיות לקבלת קשתות



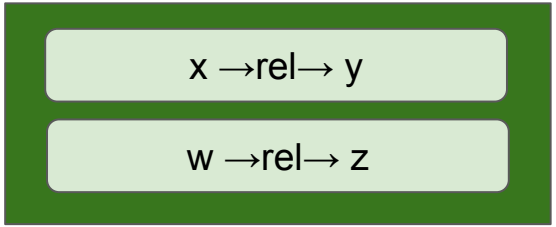
קשתות שכבר הוספנו לעץ

מעברים

- **הזח SHIFT - העברת מילה מהחיץ למחסנית**
- קשת שמאלית LEFT-ARC - הוסף קשת מהמילה העליונה במחסנית לשניה והסר את השניה
- קשת ימנית RIGHT-ARC - הוסף קשת מהמילה השניה במחסנית לעליונה והסר את העליונה

חיץ buffer - המילים מהמשפט בהן טרם טיפלנו

מחסנית stack - מועמדות מיידיות לקבלת קשתות



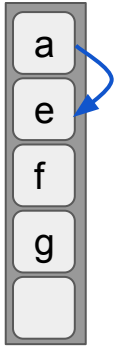
קשתות שכבר הוספנו לעץ

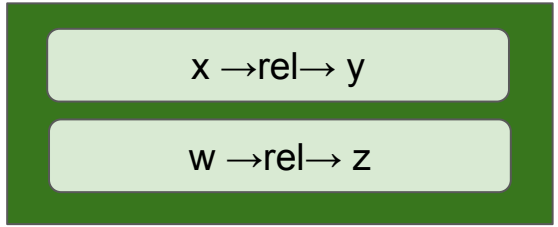
מעברים

- הזח SHIFT - העברת מילה מהחיץ למחסנית
- **קשת שמאלית LEFT-ARC** - הוסף קשת מהמילה העליונה במחסנית לשניה והסר את השניה
- קשת ימנית RIGHT-ARC - הוסף קשת מהמילה השניה במחסנית לעליונה והסר את העליונה

חיץ buffer - המילים מהמשפט בהן טרם טיפלנו

מחסנית stack - מועמדות מיידיות לקבלת קשתות





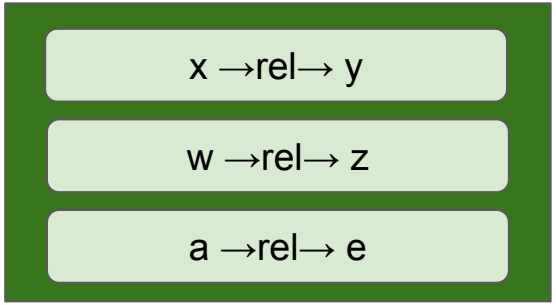
קשתות שכבר הוספנו לעץ

מעברים

- הזח SHIFT - העברת מילה מהחיץ למחסנית
- **קשת שמאלית LEFT-ARC** - הוסף קשת מהמילה העליונה במחסנית לשניה והסר את השניה
- קשת ימנית RIGHT-ARC - הוסף קשת מהמילה השניה במחסנית לעליונה והסר את העליונה

חיץ buffer - המילים מהמשפט בהן טרם טיפלנו

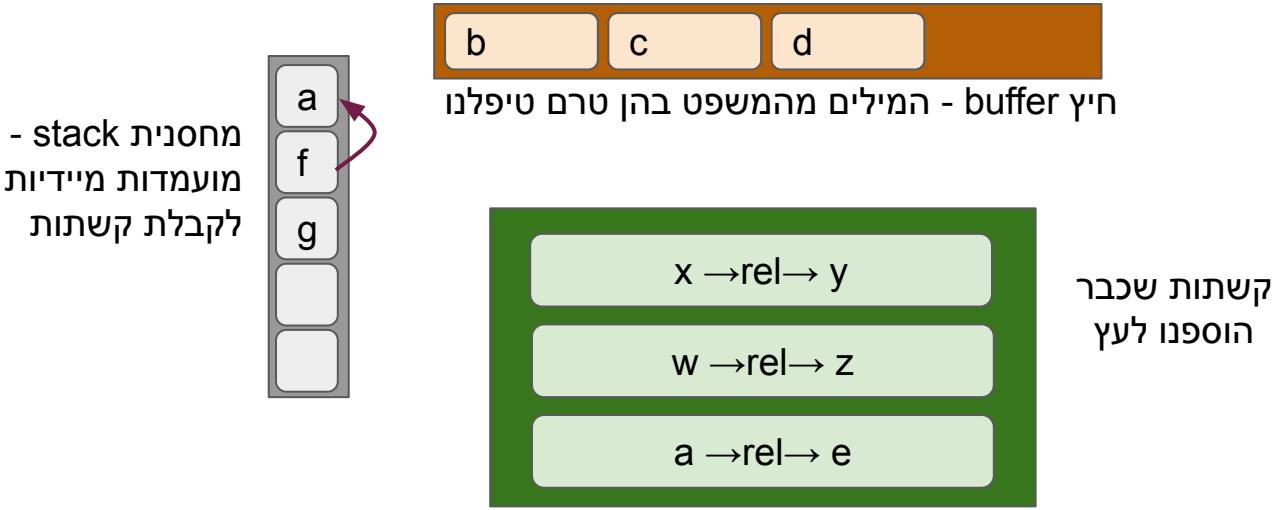
מחסנית stack - מועמדות מיידיות לקבלת קשתות



קשתות שכבר הוספנו לעץ

מעברים

- הזח SHIFT - העברת מילה מהחיץ למחסנית
- קשת שמאלית LEFT-ARC - הוסף קשת מהמילה העליונה במחסנית לשניה והסר את השניה
- **קשת ימנית RIGHT-ARC** - הוסף קשת מהמילה השניה במחסנית לעליונה והסר את העליונה

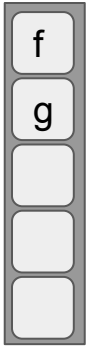


מעברים

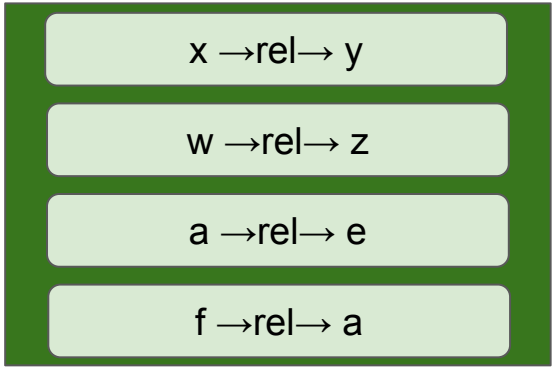
- הזח SHIFT - העברת מילה מהחיץ למחסנית
- קשת שמאלית LEFT-ARC - הוסף קשת מהמילה העליונה במחסנית לשניה והסר את השניה
- **קשת ימנית RIGHT-ARC** - הוסף קשת מהמילה השניה במחסנית לעליונה והסר את העליונה

- מצב התחלתי?
- מצב סופי?
- סיבוכיות?
- מגבלות?

חיץ buffer - המילים מהמשפט בהן טרם טיפלנו



מחסנית stack - מועמדות מיידיות לקבלת קשתות



קשתות שכבר הוספנו לעץ

הדגמה

maca’ti mixtav she katva mora le teatron

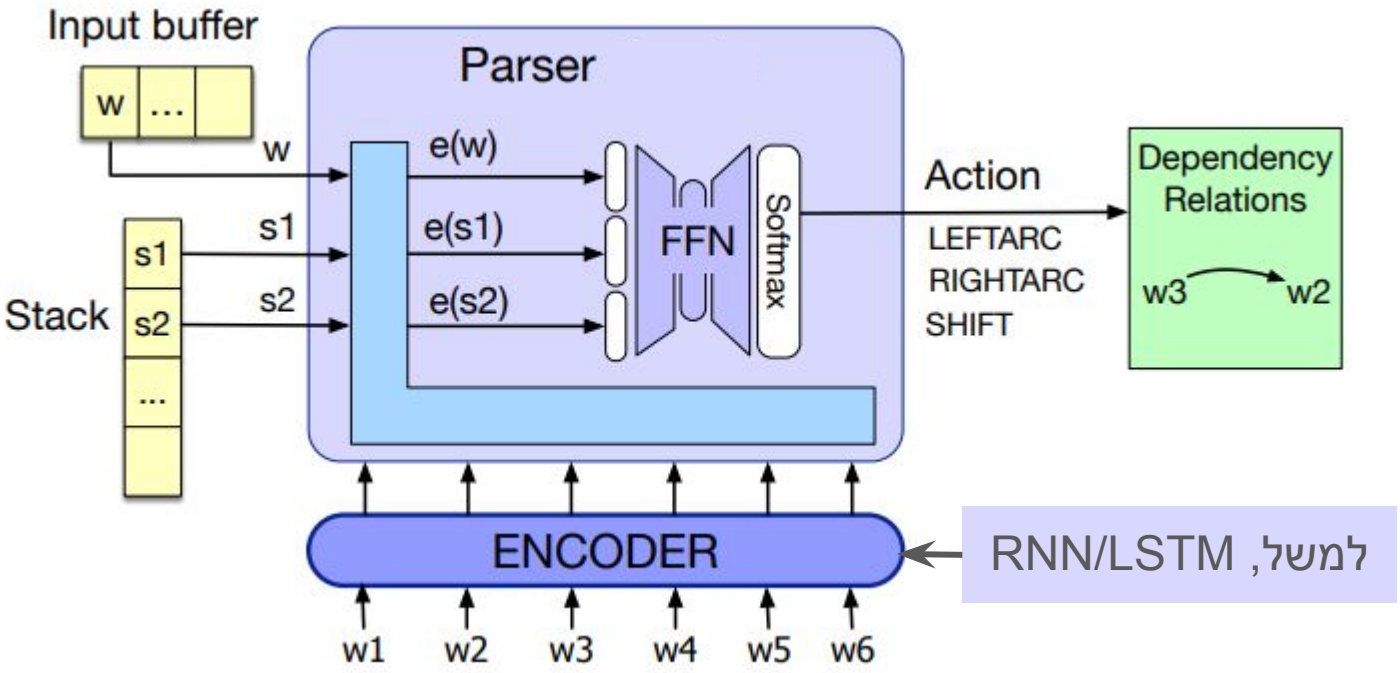
האורקל (Oracle)

- הדאטא שלנו נתון כעץ: אוסף קשתות לא סדור
- אנחנו צריכים סדרת החלטות
- הרכיב שמתרגם את הראשון לשני נקרא **אורקל** (יודע-כל)
- ניתן לבנות אורקל באופן דטרמיניסטי
  - לא ניתן להבטיח שייתן את סט המעברים **היחיד**
- (מה קורה לפונקציית ההפסד? למטריקה?)

ציונים למעברים

- כדי שהמנתח (parser) שלנו יידע איך לפעול, הוא צריך לקבל החלטה פר:
  - מצב של המערכת (חיץ, מחסנית, קשתות קיימות)
  - החלטה מועמדת (הזח, קשת-ימין, קשת-שמאל)

- זו בעיית **סיווג** שניתן לגשת אליה ע"י חילוץ פיצ'רים מתוך איברי המערכת ולתת ציון לכל פעולה
  - למשל, אם בראש המחסנית יש NOUN ובמקום השני DET, ניתן ציון גבוה לקשת-שמאל
  - פיצ'רים מקובלים - המילים במקומות העליונים במחסנית, המילים בתחילת החיץ, חלקי הדיבר שלהן, שרשור שני האלמנטים, קשתות שכבר יצאו מהמילים במחסנית ותתי-העץ בהמשך הדרך, כל שילוב בין כל המוזכרים לעיל
  - עוד אפשרות - פיצ'רים נוירוניים שכוללים הקשר מכלל המשפט (שקף הבא)



למשל, RNN/LSTM

חיפוש אלומה (beam search)

- האלגוריתם שראינו הוא מאוד "החלטי" - אין איך לחזור חזרה, אין ויטרבי, מקסימום מקומי תמיד נותן תת-עץ של העץ הסופי

- פתרון אחד - חיפוש אלומה
  - מחזיקים תמיד את ה-k תתי-העצים הכי טובים עד כה (הציונים הם סכום ציוני הקשתות)
  - בכל צעד מתייחסים לכל האלומה ומעדכנים את מה שנשאר עליה

```
function DependencyBeamParse(words, width) returns dependency tree

  state ← {[root], [words], [], 0.0}     ;initial configuration
  agenda ← ⟨state⟩          ;initial agenda

  while agenda contains non-final states
    newagenda ← ⟨⟩
    for each state ∈ agenda do
      for all {t | t ∈ ValidOperators(state)} do
        child ← Apply(t, state)
        newagenda ← AddToBeam(child, newagenda, width)
    agenda ← newagenda
  return BestOf(agenda)

function AddToBeam(state, agenda, width) returns updated agenda

  if Length(agenda) < width then
      agenda ← Insert(state, agenda)
  else if Score(state) > Score(WorstOf(agenda))
      agenda ← Remove(WorstOf(agenda))
      agenda ← Insert(state, agenda)
  return agenda
```

Figure 14.11 Beam search applied to transition-based dependency parsing.

מערכות מעברים חלופיות

- פתרון אחר לבעיית ה"החלטיות" - שימוש במערכת מעברים שממתינה עם ההחלטות שנוטות להיות פזיזות

- אנחנו למדנו Arc-standard. בניגוד אליה, Arc-eager יוצרת קשתות-ימינה ברגע שאפשר, בלי להיפטר מהמילה העליונה (פעולה שעבורה מוגדר מעבר חדש בשם Reduce)
  - האם העצים שיכולים להיווצר ע"י המערכות הללו שקולים?

- יש עוד הרבה מערכות מעברים שנוסחו לטובת בעיות אחרות שנצפו אמפירית. יש גם מערכות שנוסחו במיוחד עבור יצירת עצים לא-היטליים (nonprojective)

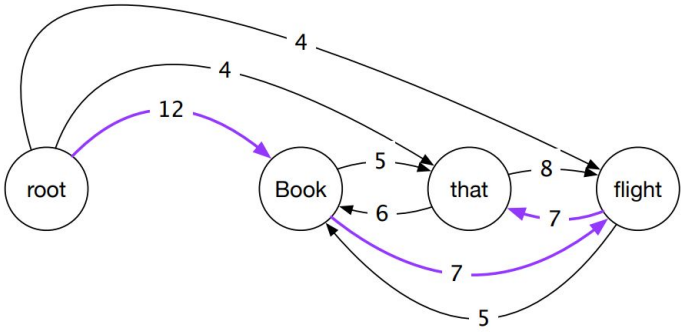
ניתוח תלויות בגרף (Graph-based dependency parsing)

- קצת כמו המגבלות של HMM בתיוג, ניתוח מבוסס-מעברים מקבל החלטות **מקומיות** שהיכולת שלהן להסתמך על כלל העץ הנוצר מוגבלת.
  - בפרט, מתקשה מאוד עם **קשתות ארוכות** (*האריה שאכל את הזברה שליחכה את העשב שצמח באחו רבץ*)
- ניתוח תלויות מבוסס-גרף מסתכל על כל **עץ שלם**, נותן לו ציון כולל, ובוחר את בעל הציון הגבוה ביותר
- כרגיל, יש לנו פה בעיית חיפוש (כמה עצים אפשריים אמרנו שיש?), אז נפעיל הנחות מקלות

נ"ת בגרף - פירוק קשתות (edge-factoring / arc-factoring)

$$\widehat{T}(X) = argmax_{t \in \mathcal{T}(X)} score(t, S) = \sum_{e \in t} score'(e)$$

- נגדיר ציון לכל קשת
  - לכל זוג צמתים
    - לשני הכיוונים האפשריים
      - לכל סוג קשת אפשרית
  - + לכל צומת כשורש
- הציון אינו תלוי בקשתות האחרות בעץ שנבחר
- (מכירות.ים אלגוריתם למקסום סכום ציונים על גרף מכוון?)



מציאת עץ פורש מקסימלי (MST)

- אלגוריתם Chu-Liu-Edmonds

- ניתן ציון לכל הקשתות האפשריות (כולל מהשורש)
- נמצא לכל צומת את הקשת המקסימלית ה**נכנסת** אליו
- יכול להיות שמצאנו מעגל
  - נפחית מכל קשת שנכנסת לצומת במעגל את הערך המקסימלי שנכנס לצומת
  - נכווץ את המעגל לצומת בודד ונקרא לאלגוריתם על העץ החדש
  - נרחיב מחדש את העץ ונשחזר את המעגל הפנימי בלי הקשת שנכנסת לצומת שקיבל את הקשת מהעץ הגדול
- קיבלנו עץ, נחזיר אותו

- (סיבוכיות?)

מתן ציונים לקשתות

- כמו קודם, אפשר לחלץ פיצ'רים וללמוד משקלות

- פיצ׳רים מועילים:
  - מילים, למות, חלקי דיבר, צורת המילה (ראשה + תלויה)
  - מילים, למות וכו׳ (לפני ואחרי הראשה והתלויה, וביניהן)
  - וקטורים משיכוני מילים
  - תג הקשת (גם אותם אפשר לשכן במרחב וקטורי)
  - כיוון הקשת
  - המרחק בין המילים

- למידה
  - ההפסד מגיע מאוחר בתהליך (כל המבנה חייב להתגלות)
  - יחידת החיזוי הבסיסית היא **ציון מספרי לקשת** (= רגרסיה)
  - כלל עדכון מתאים - פרספטרון / SVM

מערכת נוירונית למציאת גרף תלויות

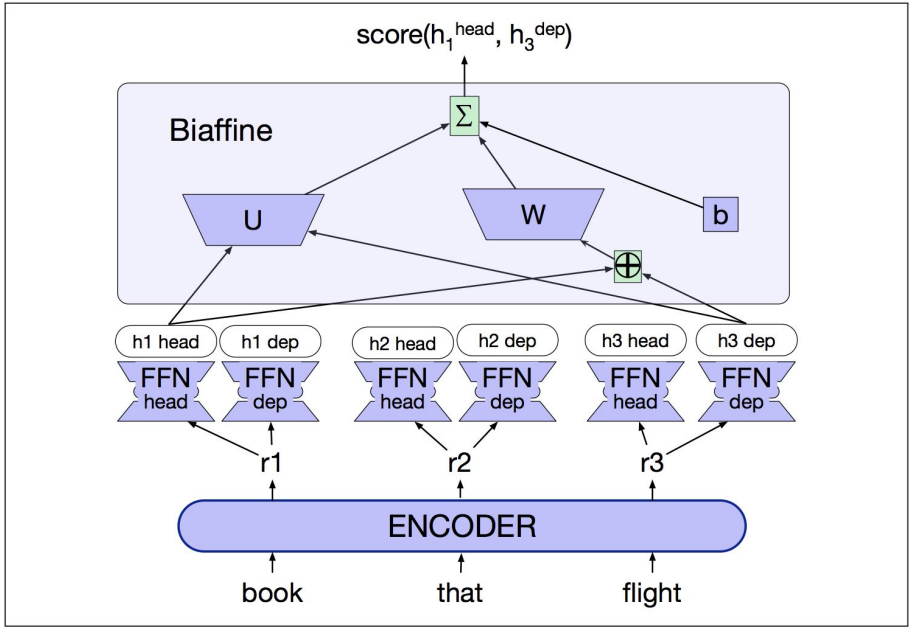


Figure 14.15 Computing scores for a single edge (book→ flight) in the biaffine parser of Dozat and Manning (2017); Dozat et al. (2017). The parser uses distinct feedforward networks to turn the encoder output for each word into a head and dependent representation for the word. The biaffine function turns the head embedding of the head and the dependent embedding of the dependent into a score for the dependency edge.

- ייצוגים וקטוריים נפרדים למילה לפי האם היא מועמדת להיות ראשה או תלויה
- רשת ביאפינית biaffine - נועדה למנף קשרים "כפליים" בין הייצוגים
- מערך נפרד (וכמעט זהה) להחלטה על התג של הקשת
  - כלומר, עוד הנחה מקלה