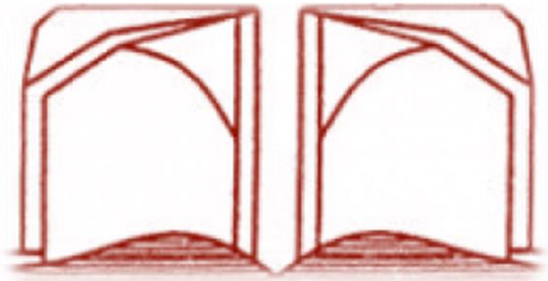


آشنایی با زبان‌شناسی رایانشی

تجزیه نحوی (Parsing)

هادی ویسی

h.veisi@ut.ac.ir

دانشگاه تهران – دانشکده علوم و فنون نوین

فهرست

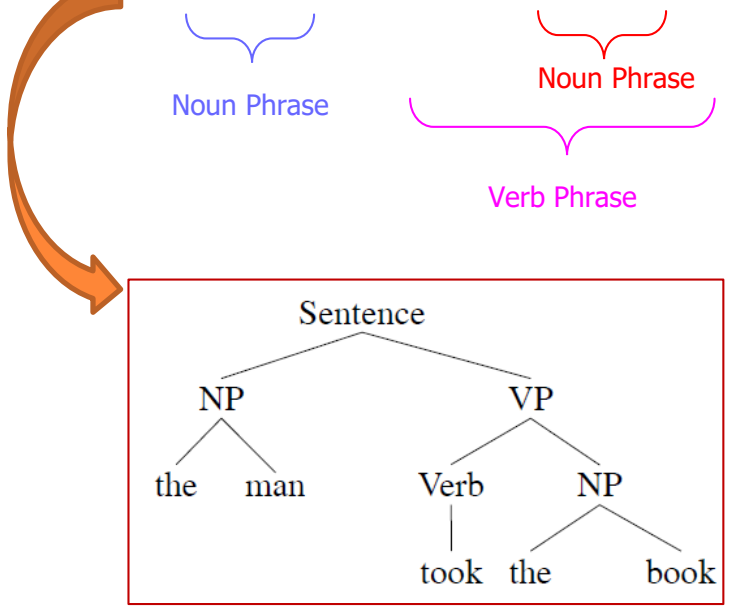
- نحو (گرامر)
  - ساختار عبارت (Phrase Structure)
  - ساختار وابستگی (Dependency Structure)
  - گرامر مستقل از متن (Context Free Grammar)
  - سلسله مراتب چامسکی (Chomsky Hierarchy)
  - Treebank
  - معیارهای مناسب بودن یک گرامر
- تجزیه نحوی (Parsing)
  - تجزیه پایین به بالا (Bottom-Up Parsing)
    - تجزیه Earley
  - تجزیه بالا به پایین (Top-Down Parsing)
    - تجزیه CKY

نحو (گرامر)

- علم مطالعه نحوه قرار گرفتن کلمات در کنار هم
  - گرامر رسمی (formal grammar) یا به اختصار «گرامر»
    - مجموعه‌ای از قواعد (rule) برای تشکیل دنباله‌های کلمات در زبان‌های رسمی (formal languages)
- واژه syntax (نحو)
  - از واژه یونانی *sýntaxis* به معنی «setting out together or arrangement»
- دارای سابقه طولانی
  - تدوین دستور زبان برای زبان سانسکریت توسط Panini در قرن ۴ پیش از میلاد
- رویکردها در ساختاردهی نحوی
  - ساختار عبارت (Phrase Structure) - اجزای تشکیل‌دهنده (Constituency)
  - ساختار وابستگی (Dependency Structure)

نحو: ساختار عبارت (Phrase Structure) . . .

- سازمان‌دهی کلمات در عباراتی تو در تو
  - مثال: جمله «the man took the book»



VP → … VB* …
NP → … NN* …
ADJP → … JJ* …
ADVP → … RB* …

- به هر کدام از واحدهای فوق یک جز تشکیل‌دهنده (Constituent) گفته می‌شود

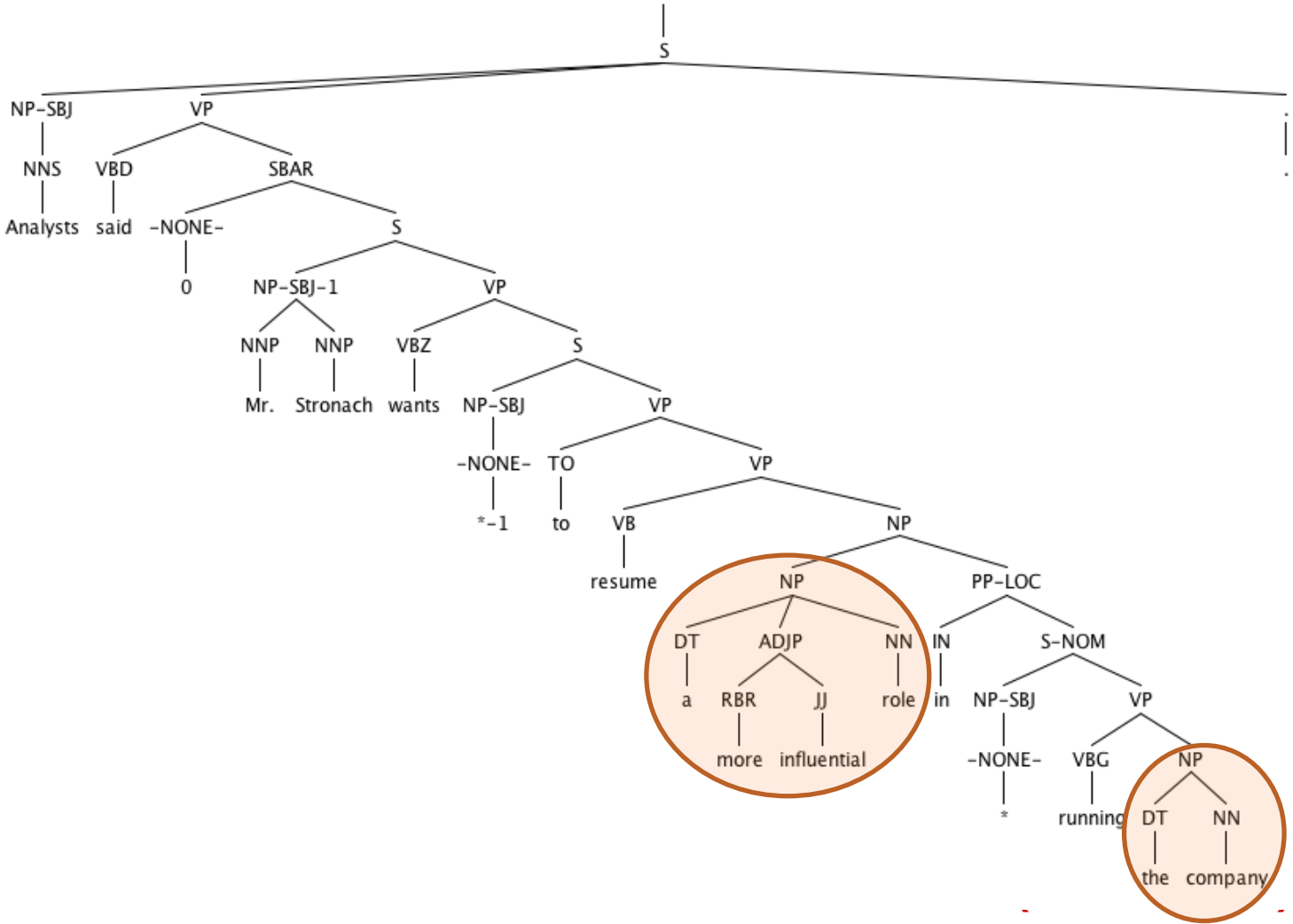
نحو: ساختار عبارت (Phrase Structure) . . .

- نحوه شناسایی جز تشکیل‌دهنده (Constituent)
  - کار آسانی نیست و در مواردی توافق همگانی وجود ندارد
  - ظهور در بخش‌هایی از جمله با ساختار نحوی مشابه (مثال: قبل از فعل)
    - [three parties from Brooklyn] arrive. . .
    - [a high-class spot such as Mindy's] attracts. . .
    - [the Broadway coppers] love. . .
    - [they] sit
  - توزیع: یک جز تشکیل‌دهنده به عنوان یک واحد رفتار کرده و می تواند در جاهایی مختلفی از جمله بیاید
    - John talked [to the children] [about drugs]
    - John talked [about drugs] [to the children]
    - [Incorrect] John talked drugs to the children about
  - جایگزینی: به جای هم به کار می‌روند
    - I sat [on the box/right on top of the box/there]
  - معنی/ ساختار درونی منظم/تناسب/...

نحو: ساختار عبارت (Phrase Structure)

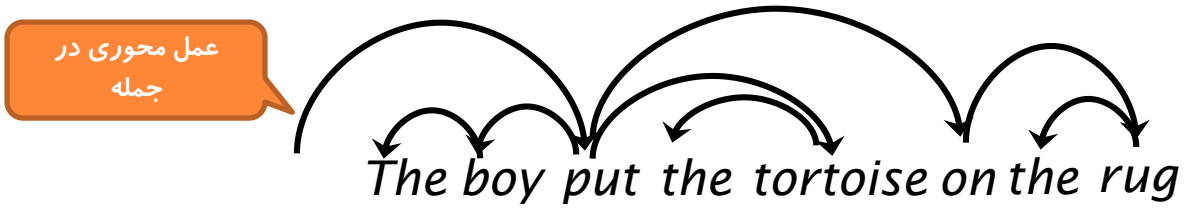
- **مثال**

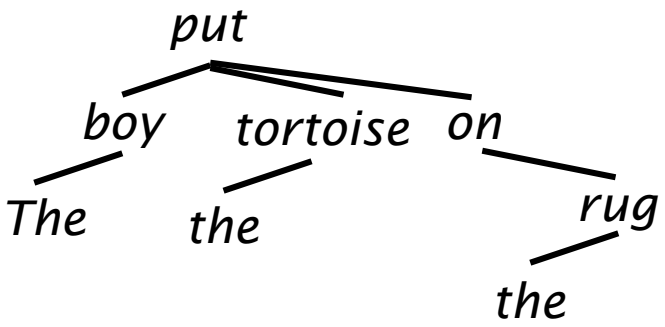
Analysts said Mr. Stronach wants to resume a more inflectional role in running the company



نحو: ساختار وابستگی (Dependency Structure)

- بیانگر این است که کدام کلمه به کدام کلمه دیگر وابسته است





گرامر مستقل از متن (Context Free Grammar) . . .

- گرامر مستقل از متن
  - CFG: Context Free Grammar
    - نام دیگر: Phrase-Structure Grammars
  - نمایش ساختار با BNF: Backus-Naur Form
  - ارائه ساختار رسمی توسط چامسکی (۱۹۵۶)

  - تشکیل شده از
    - مجموعه‌ای واژگان (حاوی کلمات یا علائم)
    - مجموعه‌ای از قوانین که بیانگر نحوه گروه‌بندی کلمات (علائم) هستند

- استفاده در توصیف زبان‌های طبیعی و زبان‌های برنامه‌نویسی
  - به اندازه کافی قوی است که بتوان ساختارهای زبان طبیعی را با آن نمایش داد.
  - به اندازه کافی محدود هست که بتوان تجزیه‌گرهای موثری را برای تجزیه جملات بر اساس آن طراحی کرد.

گرامر مستقل از متن . . .

- مثال: عبارت اسمی (NP)
  - یا اسم خاص (ProperNoun) است
  - یا به صورت حرف تعریف (Det) که بعد از آن Nominal می‌آید، است

$$NP \rightarrow ProperNoun$$
$$NP \rightarrow Det\ Nominal$$

  - خود Nominal می‌تواند اسم (Noun) باشد یا اسم (Nominal Noun)

$$Nominal \rightarrow Noun \mid Nominal\ Noun$$

یا

  - هرکدام از عناوین (علائم) می‌تواند شامل برخی از کلمات باشند

$Det \rightarrow a$ $Det \rightarrow the$ $Noun \rightarrow flight$

$Noun \rightarrow book$ $Nominal\ Noun \rightarrow cup\ of\ tea$ $ProperNoun \rightarrow Ali\ Karimi$

- بنابراین

$NP \rightarrow a\ flight$ $NP \rightarrow the\ cup\ of\ tea$ $NP \rightarrow Ali\ Karimi$

گرامر مستقل از متن . . .

- استفاده از یک گرامر مستقل از متن

$NP \rightarrow ProperNoun$
$NP \rightarrow Det\ Nominal$
$Nominal \rightarrow Noun \mid Nominal\ Noun$

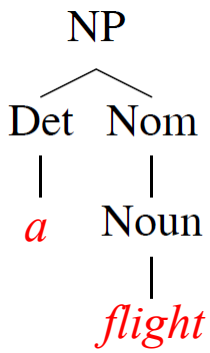
$Det \rightarrow a$
$Det \rightarrow the$
$Noun \rightarrow flight$

$Noun \rightarrow book$
$Nominal\ Noun \rightarrow cup\ of\ tea$
$ProperNoun \rightarrow Ali\ Karimi$

- تولید (مشتق شدن) دنباله‌ای از رشته‌ها (کلمات) با استفاده از قوانین گرامر
  - تشخیص (recognition) پذیرش یک رشته توسط گرامر

$NP \rightarrow a\ flight$

- انتساب یک ساختار به یک جمله (دنباله‌ای از کلمات)
  - درخت تجزیه (parse tree)



گرامر مستقل از متن . . .

- **عناصر یک گرامر مستقل از متن**
  - **پایانه** (terminal): **عناصری که متناسب با یک واژه در زبان هستند**
    - از پایانه‌های انشعاب جدیدی ایجاد نمی‌شود
    - سمت راست →
    - مانند the، book، Ali و ...
  - **غیرپایانه** (non-terminal)
    - علائمی که بیانگر یک خوشه یا توسعه‌ای از آنها هستند
    - سمت چپ (فقط یک غیرپایانه) یا راست →

غیرپایانه

$$NP \rightarrow ProperNoun$$
$$NP \rightarrow Det\ Nominal$$
$$Nominal \rightarrow Noun \mid Nominal\ Noun$$

پایانه

$$Det \rightarrow a$$
$$Det \rightarrow the$$
$$Noun \rightarrow flight$$

گرامر مستقل از متن . . .

- گرامر انگلیسی . . .

شروع (sentence)

عبارت اسمی

$S \rightarrow NP\ VP$ I prefer a morning flight

$VP \rightarrow Verb\ NP$ prefer a morning flight

عبارت حرف اضافه

$VP \rightarrow Verb\ NP\ PP$ leave Boston in the morning

$VP \rightarrow Verb\ PP$ leaving on Thursday

$PP \rightarrow Preposition\ NP$ from Los Angeles

گرامر مستقل از متن . . .

- گرامر انگلیسی . . .

قوانین گرامر

$S \rightarrow NP\ VP$ I + want a morning flight

$NP \rightarrow Pronoun$ I
| $Proper\text{-}Noun$ Los Angeles
| $Det\ Nominal$ a + flight

$Nominal \rightarrow Nominal\ Noun$ morning + flight
| $Noun$ flights

$VP \rightarrow Verb$ do
| $Verb\ NP$ want + a flight
| $Verb\ NP\ PP$ leave + Boston + in the morning
| $Verb\ PP$ leaving + on Thursday

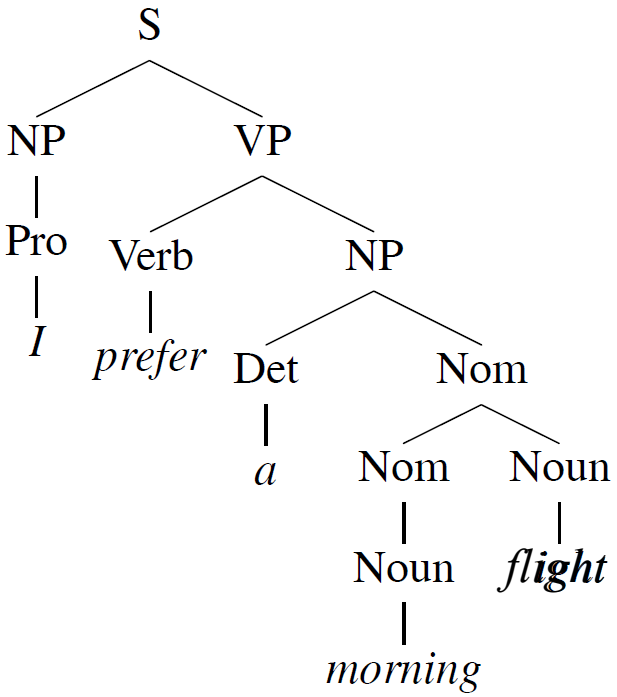
$PP \rightarrow Preposition\ NP$ from + Los Angeles

واژگان گرامر

Noun	→	*flights* | *breeze* | *trip* | *morning* | . . .
Verb	→	*is* | *prefer* | *like* | *need* | *want* | *fly*
Adjective	→	*cheapest* | *non – stop* | *first* | *latest* | *other* | *direct* | . . .
Pronoun	→	*me* | *I* | *you* | *it* | . . .
Proper-Noun	→	*Alaska* | *Baltimore* | *Los Angeles* | *Chicago* | *United* | *American* | . . .
Determiner	→	*the* | *a* | *an* | *this* | *these* | *that* | . . .
Preposition	→	*from* | *to* | *on* | *near* | . . .
Conjunction	→	*and* | *or* | *but* | . . .

گرامر مستقل از متن . . .

- گرامر انگلیسی
  - درخت تجزیه برای جمله «I prefer a morning flight»



  - نمایش کروشه‌ای (bracketed notation)

[*S* [*NP* [*Pro* I]] [*VP* [*V* prefer] [*NP* [*Det* a] [*Nom* [*N* morning] [*Nom* [*N* flight]]]]]]

گرامر مستقل از متن

- تعریف رسمی یک گرامر مانند G
  - شامل چهار جز به صورت $G=(N,\Sigma,R,S)$
  - N: مجموعه‌ای متناهی از غیرپایانه‌ها (nonterminal symbols)
    - غیرپایانه‌ها یا متغیرها (variables)
  - Σ: مجموعه‌ای متناهی از پایانه‌ها (terminal symbols)
    - متمایز از مجموعه N
    - پایانه‌ها کلمات زبان را مشخص می‌سازند
  - R: مجموعه‌ای از قواعد به شکل $A\rightarrow\beta$
    - A مجموعه‌ای از غیرپایانه‌ها (حداقل یک مورد)
      - $|LHS|=1$ (سمت چپ یک واحد دارد)
    - β دنباله‌ای از پایانه‌ها و غیرپایانه‌ها است از $(\Sigma\cup N)^*$ – می‌تواند تهی یا بینهایت باشد
  - S: یکی از غیرپایانه‌ها به عنوان نماد آغازین

صورت نرمال چامسکی (CNF: Chomsky Normal Form)

- حالتی از گرامر مستقل از متن (CFG) که سمت راست آن
  - فقط یک پایانه است $A \rightarrow a$
  - یا دو غیرپایانه است $A \rightarrow B\ C$
- یادآوری
  - گرامر CFG در سمت راست می‌تواند هر تعداد علامت داشته باشد
- یعنی، گرامر در سمت راست
  - دارای حالت ترکیبی پایانه و غیرپایانه نیست $NP \rightarrow the\ NN$
  - دارای تولید واحد (unit production) برای غیرپایانه‌ها نیست $NP \rightarrow NNP$
  - دارای بیش از دو غیرپایانه نیست $VP \rightarrow VPZ\ NP\ PP$

صورت نرمال چامسکی (CNF) . . .

- هر گرامر CFG می‌تواند به یک CNF تبدیل شود (معادل ضعیف)
  - معادل ضعیف (Weak equivalence): دو گرامر را گویند که هر دو جملات یکسانی را تولید می‌کنند.
    - معادل قوی (Strong equivalence): دو گرامر را گویند که هر دو جملات یکسانی را تولید می‌کنند و همچنین ساختار یکسانی را برای آن جملات ایجاد می‌کنند.

- مراحل تبدیل CFG به CNF
  - حذف پایانه‌ها از حالت ترکیبی ($NP \rightarrow the\ NN$)
  - حذف تولید واحدها (unit productions) ($NP \rightarrow NNP$)
  - حذف قوانین دارای بیش از دو غیرپایانه ($VP \rightarrow VPZ\ NP\ PP$) (دودویی کردن)

صورت نرمال چامسکی: تبدیل CFG به CNF ...

- حذف پایانه‌ها از حالت ترکیبی ($NP \rightarrow the\ NN$)

- **foreach** production $A \rightarrow B_1\ B_2 \ldots B_k$, containing a terminal x
- Add new non-terminal/production $X1 \rightarrow x$ (unless it has already been added)
- Replace every $B_i = x$ with $X1$

$INF\text{-}VP \rightarrow to\ VP$

$INF\text{-}VP \rightarrow TO\ VP$
$TO \rightarrow to$

- حذف تولید واحدها (unit productions)

- while there is a unit-production $A \rightarrow B$,
- Remove $A \rightarrow B$.
- **foreach** $B \rightarrow u$, add $A \rightarrow u$.

- حذف قوانین دارای بیش از دو غیرپایانه (دودویی کردن)

- **foreach** rule p of form $A \rightarrow B_1\ B_2 \ldots B_k$
- replace p with $A \rightarrow B_1\ X1$, $X1 \rightarrow B_2\ X2$, $X2 \rightarrow B_3\ X3$, ..., $X(k-2) \rightarrow B_{k-1}\ B_k$ (Xi's are new variables.)

$A \rightarrow B\ C\ \gamma$

$X1 \rightarrow B\ C$
$A \rightarrow X1\ \gamma$

صورت نرمال چامسکی: تبدیل CFG به CNF ...

- مثال

$S \rightarrow NP\ VP$	$S \rightarrow NP\ VP$
$S \rightarrow Aux\ NP\ VP$	$S \rightarrow X1\ VP$
	$X1 \rightarrow Aux\ NP$
$S \rightarrow VP$	$S \rightarrow book \mid include \mid prefer$
	$S \rightarrow Verb\ NP$
	$S \rightarrow X2\ PP$
	$S \rightarrow Verb\ PP$
	$S \rightarrow VP\ PP$
$NP \rightarrow Pronoun$	$NP \rightarrow I \mid she \mid me$
$NP \rightarrow Proper\text{-}Noun$	$NP \rightarrow TWA \mid Houston$
$NP \rightarrow Det\ Nominal$	$NP \rightarrow Det\ Nominal$
$Nominal \rightarrow Noun$	$Nominal \rightarrow book \mid ight \mid meal \mid money$
$Nominal \rightarrow Nominal\ Noun$	$Nominal \rightarrow Nominal\ Noun$
$Nominal \rightarrow Nominal\ PP$	$Nominal \rightarrow Nominal\ PP$
$VP \rightarrow Verb$	$VP \rightarrow book \mid include \mid prefer$
$VP \rightarrow Verb\ NP$	$VP \rightarrow Verb\ NP$
$VP \rightarrow Verb\ NP\ PP$	$VP \rightarrow X2\ PP$
	$X2 \rightarrow Verb\ NP$
$VP \rightarrow Verb\ PP$	$VP \rightarrow Verb\ PP$
$VP \rightarrow VP\ PP$	$VP \rightarrow VP\ PP$
$PP \rightarrow Preposition\ NP$	$PP \rightarrow Preposition\ NP$

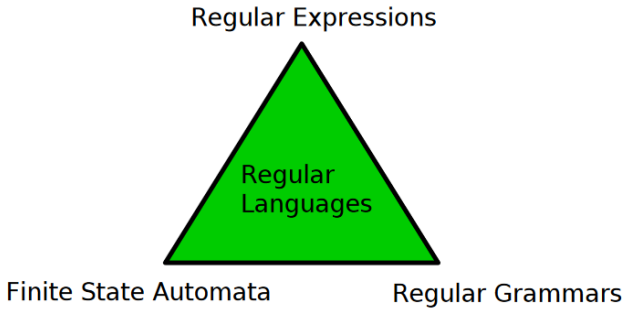
صورت نرمال چامسکی (CNF)

- تولید درخت تجزیه دودویی (binary branching)
  - در هر مرحله از ساخت درخت تجزیه، دو شاخه وجود دارد
  - استفاده در الگوریتم تجزیه CKY

VP → ADV eat NP

VP → ADVP VP_1
VP_1 → VP_2 NP
VP_2 → eat

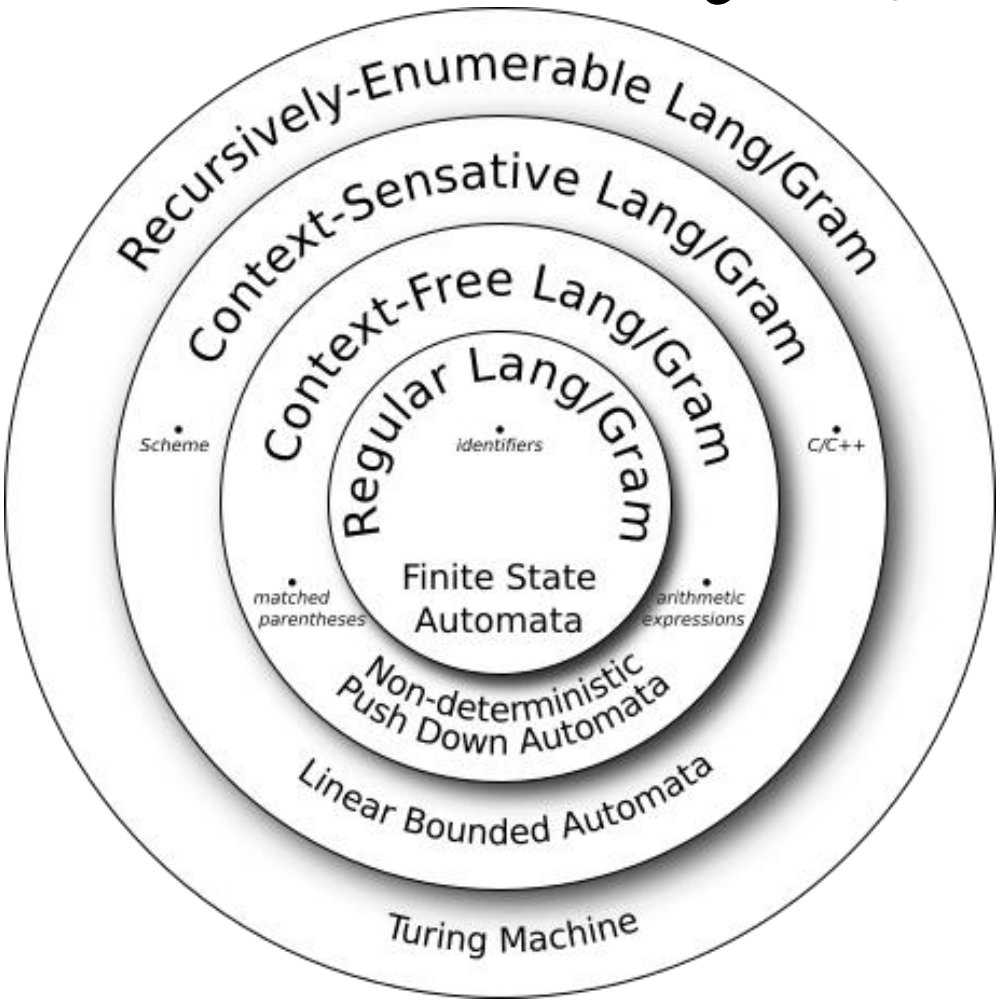
سلسله مراتب چامسکی (Chomsky Hierarchy) ...



- دسته‌بندی گرامرهای رسمی در ۴ سطح
  - **گرامرهای نوع ۳** (regular grammars)
    - قواعد به صورت‌های $A \rightarrow aB$، $A \rightarrow Ba$ یا $A \rightarrow a$ هستند
      - A و B غیرپایانه و a پایانه است.
      - $A \rightarrow \varepsilon$ نیز مجاز می باشد.
  - **گرامرهای نوع ۲** (context free grammars)
    - قواعد به صورت $A \rightarrow \beta$ می‌باشد که A یک غیرپایانه و β دنباله‌ای از پایانه‌ها و غیرپایانه‌ها است.
  - **گرامرهای نوع ۱** (context sensitive grammars)
    - قواعد به صورت $\alpha A \beta \rightarrow \alpha \gamma \beta$ می‌باشد که A غیرپایانه و α، β، γ دنباله‌ای از پایانه‌ها و غیرپایانه‌هاست
    - α و β می‌توانند تهی باشند ولی γ باید حداقل شامل یک نماد باشد.
      - $|LHS| \leq |RHS|$
  - **گرامرهای نوع ۰** (unrestricted grammars) یا (recursively enumerable languages)
    - محدودیتی برروی قواعد اعمال نمی‌گردد (ماشین تورینگ)

سلسله مراتب چامسکی (Chomsky Hierarchy)

- دسته‌بندی گرامرهای رسمی در ۴ سطح



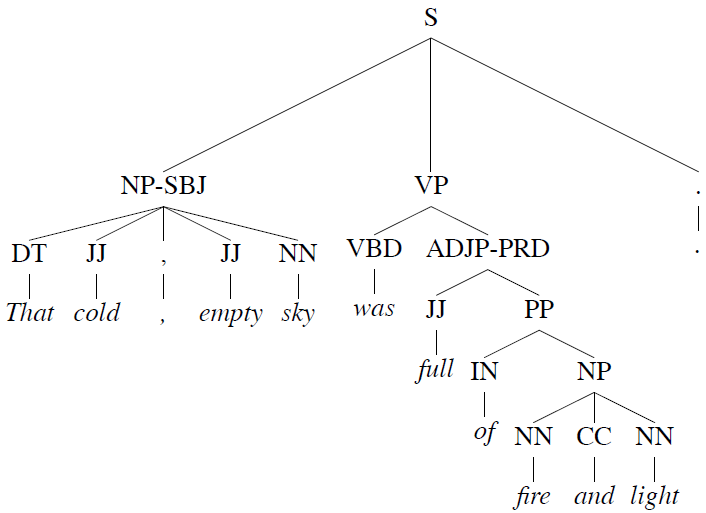
معیارهای مناسب بودن یک گرامر

- عمومیت (Generality)
  - محدودهٔ جملات مجاز زبان که بر اساس گرامر به درستی تحلیل می‌شوند (پذیرفته می‌شوند).
- گزینندگی (Selectivity)
  - محدودهٔ جملات غیرمجاز در زبان که بر اساس گرامر رد می‌شوند.
- قابلیت فهم (Understandability)
  - سادگی گرامر

... Treebank

- پیکره‌ای متنی که جملات آن تجزیه نحوی شده است
  - پروژه Penn Treebank
    - برای زبان‌های انگلیسی، عربی و چینی
    - انگلیسی: شامل جملات پیکره‌های Brown، Switchboard، ATIS و Wall Street Journal

```
((S
   (NP-SBJ (DT That)
     (JJ cold) (, ,)
     (JJ empty) (NN sky) )
   (VP (VBD was)
     (ADJP-PRD (JJ full)
       (PP (IN of)
         (NP (NN fire)
           (CC and)
           (NN light) ))))
   (. .) ))
```



Treebank

```
((S
  (NP-SBJ (DT That)
    (JJ cold) (, ,)
    (JJ empty) (NN sky) )
  (VP (VBD was)
    (ADJP-PRD (JJ full)
      (PP (IN of)
        (NP (NN fire)
          (CC and)
          (NN light) ))))
  (. .) ))
```

```
((S
   (NP-SBJ The/DT flight/NN )
   (VP should/MD
     (VP arrive/VB
       (PP-TMP at/IN
         (NP eleven/CD a.m/RB ))
       (NP-TMP tomorrow/NN ))))))
```

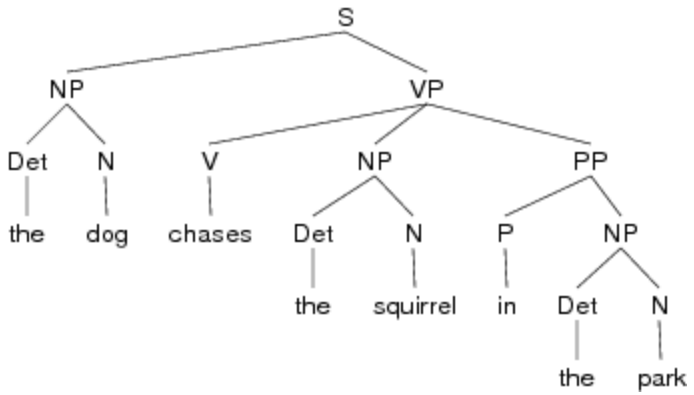
```
( (S (`` ``)
    (S-TPC-2
      (NP-SBJ-1 (PRP We) )
      (VP (MD would)
        (VP (VB have)
          (S
            (NP-SBJ (-NONE- *-1) )
            (VP (TO to)
              (VP (VB wait)
                (SBAR-TMP (IN until)
                  (S
                    (NP-SBJ (PRP we) )
                    (VP (VBP have)
                      (VP (VBN collected)
                        (PP-CLR (IN on)
                          (NP (DT those) (NNS assets) ))))))))))))
    (, ,) ('' '')
    (NP-SBJ (PRP he) )
    (VP (VBD said)
      (S (-NONE- *T*-2) ))
    (. .) ))
```

استخراج گرامر از Treebank

S	→	*NP VP .*	*PRP*	→	*we* | *he*
		NP VP	*DT*	→	*the* | *that* | *those*
		" S " , NP VP .	*JJ*	→	*cold* | *empty* | *full*
		-NONE-	*NN*	→	*sky* | *fire* | *light* | *ight*
		DT NN	*NNS*	→	*assets*
		DT NN NNS	*CC*	→	*and*
		NN CC NN	*IN*	→	*of* | *at* | *until* | *on*
		CD RB	*CD*	→	*eleven*
NP	→	*DT JJ , JJ NN*	*RB*	→	*a.m*
		PRP	*VB*	→	*arrive* | *have* | *wait*
		-NONE-	*VBD*	→	*said*
VP	→	*MD VP*	*VBP*	→	*have*
		VBD ADJP	*VBN*	→	*collected*
		VBD S	*MD*	→	*should* | *would*
		VB PP	*TO*	→	*to*
		VB S			
		VB SBAR			
		VBP VP			
		VBN VP			
		TO VP			
SBAR	→	*IN S*			
ADJP	→	*JJ PP*			
PP	→	*IN NP*			

تجزیه نحوی (Parsing) . . .

- **هدف**
  - تجزیهٔ جمله به ساختارهای نحوی تشکیل دهندهٔ آن
- **نیازمندی‌ها**
  - گرامر: دربردارندهٔ توصیف رسمی از ساختارهای مجاز زبان
  - روش تجزیه: روش آنالیز جمله برای مشخص کردن ساختارهای نحوی آن بر اساس گرامر زبان
- **مثال**
  - **جمله** «The dog chases the squirrel in the park»



تجزیه نحوی: کاربردها . . .

- ترجمه ماشینی
- بررسی صحت گرامر
  - مایکروسافت ورد
- بازیابی اطلاعات
- بازشناسی گفتار
- سنتز گفتار

تجزیه نحوی . . .

- **دو روش اصلی تجزیه نحوی**
  - **تجزیه بالا به پایین** (Top-Down Parsing)
    - از نماد نماد آغازین (S) شروع می‌کنیم (بالا)
    - تمام راه‌هایی را که می‌توان نمادها را بر اساس قواعد بازنویسی کرد بررسی می‌کنیم تا جمله موردنظر تولید شود و یا تمام بازنویسی‌های ممکن بررسی شود (پایین)
    - اگر تعداد قواعد انتخابی برای هر گروه نحوی (phrase) زیاد باشد، الگوریتم پیچیده می‌شود.
  - **تجزیه پایین به بالا** (Bottom-Up Parsing)
    - از کلمات جمله شروع می کنیم (پایین)
    - براساس قواعد گرامر، نمادها را کاهش می‌دهیم تا زمانی که تنها به نماد آغازین (S) برسیم و یا دیگر نتوانیم نمادها را کاهش دهیم (بالا)
    - اگر تعداد نمادهای انتخابی برای هر کلمه زیاد باشد، الگوریتم پیچیده می‌شود.

تجزیه بالا به پایین (Top-Down Parsing) . . .

- مثال . . .
  - گرامر

$S \rightarrow NP\ VP$
$S \rightarrow Aux\ NP\ VP$
$S \rightarrow VP$
$NP \rightarrow Pronoun$
$NP \rightarrow Proper\text{-}Noun$
$NP \rightarrow Det\ Nominal$
$Nominal \rightarrow Noun$
$Nominal \rightarrow Nominal\ Noun$
$Nominal \rightarrow Nominal\ PP$
$VP \rightarrow Verb$
$VP \rightarrow Verb\ NP$
$VP \rightarrow Verb\ NP\ PP$
$VP \rightarrow Verb\ PP$
$VP \rightarrow VP\ PP$
$PP \rightarrow Preposition\ NP$

$Det \rightarrow that \mid this \mid a$
$Noun \rightarrow book \mid flight \mid meal \mid money$
$Verb \rightarrow book \mid include \mid prefer$
$Pronoun \rightarrow I \mid she \mid me$
$Proper\text{-}Noun \rightarrow Houston \mid TWA$
$Aux \rightarrow does$
$Preposition \rightarrow from \mid to \mid on \mid near \mid through$

- جمله ورودی: *Book that flight*

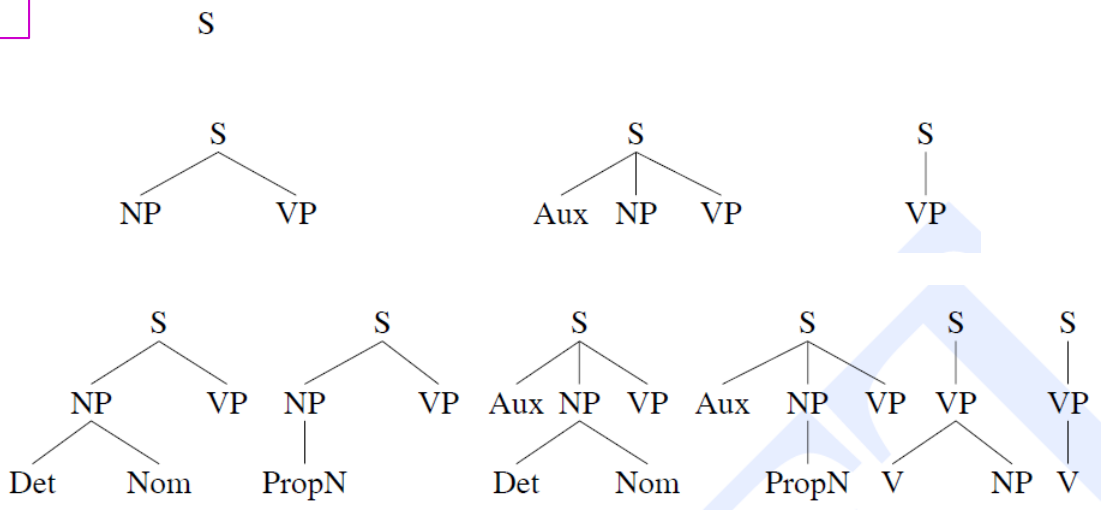
تجزیه بالا به پایین (Top-Down Parsing)

S → NP VP
S → Aux NP VP
S → VP
NP → Pronoun
NP → Proper-Noun
NP → Det Nominal
Nominal → Noun
Nominal → Nominal Noun
Nominal → Nominal PP
VP → Verb
VP → Verb NP
VP → Verb NP PP
VP → Verb PP
VP → VP PP
PP → Preposition NP

Det → that | this | a
Noun → book | flight | meal | money
Verb → book | include | prefer
Pronoun → I | she | me
Proper-Noun → Houston | TWA
Aux → does
Preposition →from | to | on | near | through

- **مثال**
  - جمله: *Book that flight*

- شروع از S
- ایجاد تمام درخت‌های شروع شده با S
- ادامه ساخت درخت بر اساس قوانین
- ...



تجزیه پایین به بالا (Bottom-Up Parsing)

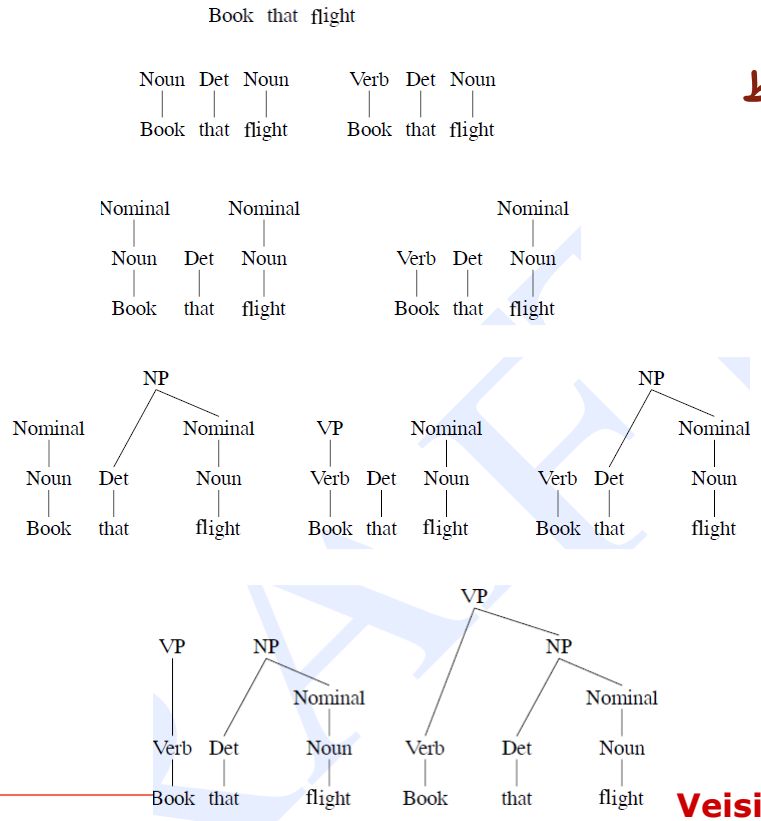
$S \rightarrow NP\ VP$
$S \rightarrow Aux\ NP\ VP$
$S \rightarrow VP$
$NP \rightarrow Pronoun$
$NP \rightarrow Proper\text{-}Noun$
$NP \rightarrow Det\ Nominal$
$Nominal \rightarrow Noun$
$Nominal \rightarrow Nominal\ Noun$
$Nominal \rightarrow Nominal\ PP$
$VP \rightarrow Verb$
$VP \rightarrow Verb\ NP$
$VP \rightarrow Verb\ NP\ PP$
$VP \rightarrow Verb\ PP$
$VP \rightarrow VP\ PP$
$PP \rightarrow Preposition\ NP$

$Det \rightarrow that \mid this \mid a$
$Noun \rightarrow book \mid flight \mid meal \mid money$
$Verb \rightarrow book \mid include \mid prefer$
$Pronoun \rightarrow I \mid she \mid me$
$Proper\text{-}Noun \rightarrow Houston \mid TWA$
$Aux \rightarrow does$
$Preposition \rightarrow from \mid to \mid on \mid near \mid through$

مثال

- جمله: *Book that flight*

- شروع از کلمات ورودی
- اعمال قوانین گرامر مرتبط
- ادامه اعمال قوانین



تجزیه پایین به بالا - تجزیه بالا به پایین

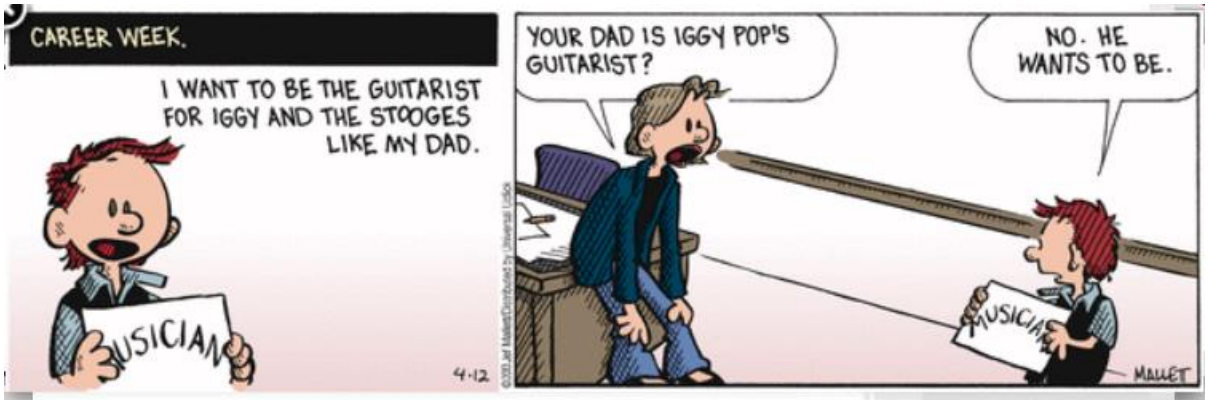
- تجزیه بالا به پایین
  - شروع با S و تولید درخت‌های (زیردرخت‌هایی) که ریشه درست دارند
  - ممکن است درخت تولید شده در ریشه با کلمات سازگار نباشد
- تجزیه پایین به بالا
  - شروع از کلمات و تولید درخت‌هایی که با جمله ورودی سازگار هستند
  - ممکن است درخت‌های تولید شده به ریشه نرسند

تجزیه نحوی . . .

- **ابهام ساختاری (Structural Ambiguity)**
  - **ابهام الصاقی (Attachment Ambiguity): یک جز می‌تواند به بیش از یک مکان در درخت تجزیه الصاق شود**
    - به ویژه در عبارات حروف اضافه (prepositional-phrase)
    - مثال I saw the man with the telescope (فارسی: من یک مرد را با تلسکوپ دیدم)
      - I [saw [the man [with the telescope]]]
        - الصاق به فاعل: with the telescope به man بر می‌گردد (مردی همراه با تلسکوپ)
      - I [saw the man with the telescope]
        - الصاق به فعل: with the telescope به saw بر می‌گردد (دیده شدن مرد به وسیله تلسکوپ)
  - **ابهام همپایگی (Coordination Ambiguity): ابهام ایجاد توسط حروف ربط**
    - مثال old men and women
      - [old [men and women]]
      - [old men] and [women]

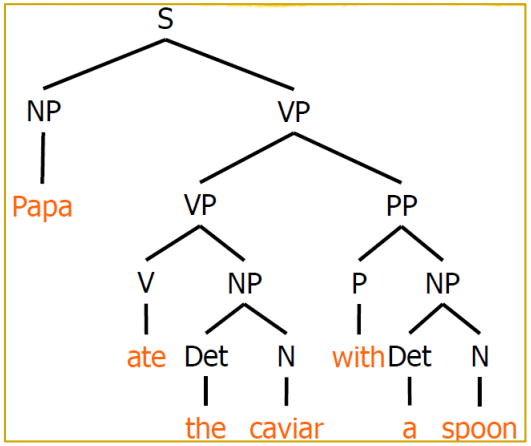
تجزیه نحوی . . .

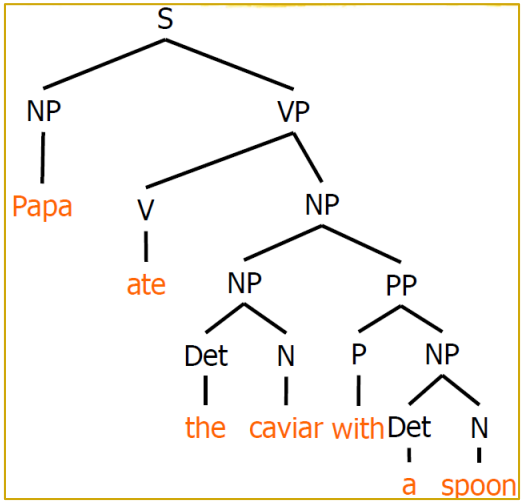
- ابهام
  - مثال



S → NP VP
NP → Det N
NP → NP PP
VP → V NP
VP → VP PP
PP → P NP

NP → Papa
N → caviar
N → spoon
V → spoon
V → ate
P → with
Det → the
Det → a





تجزیه نحوی . . .

- ابهام محلی (Local Ambiguity)
  - ابهام مربوط به اجزای جمله است (برخی کلمات) و نه کل جمله
  - ممکن است جمله فقط یک درخت تجزیه داشته باشد

- مثال: Book that flight
  - کل جمله یک درخت تجزیه دارد (عدم ابهام)
  - اما در شروع تجزیه کلمه Book می تواند فعل یا اسم باشد (ابهام)

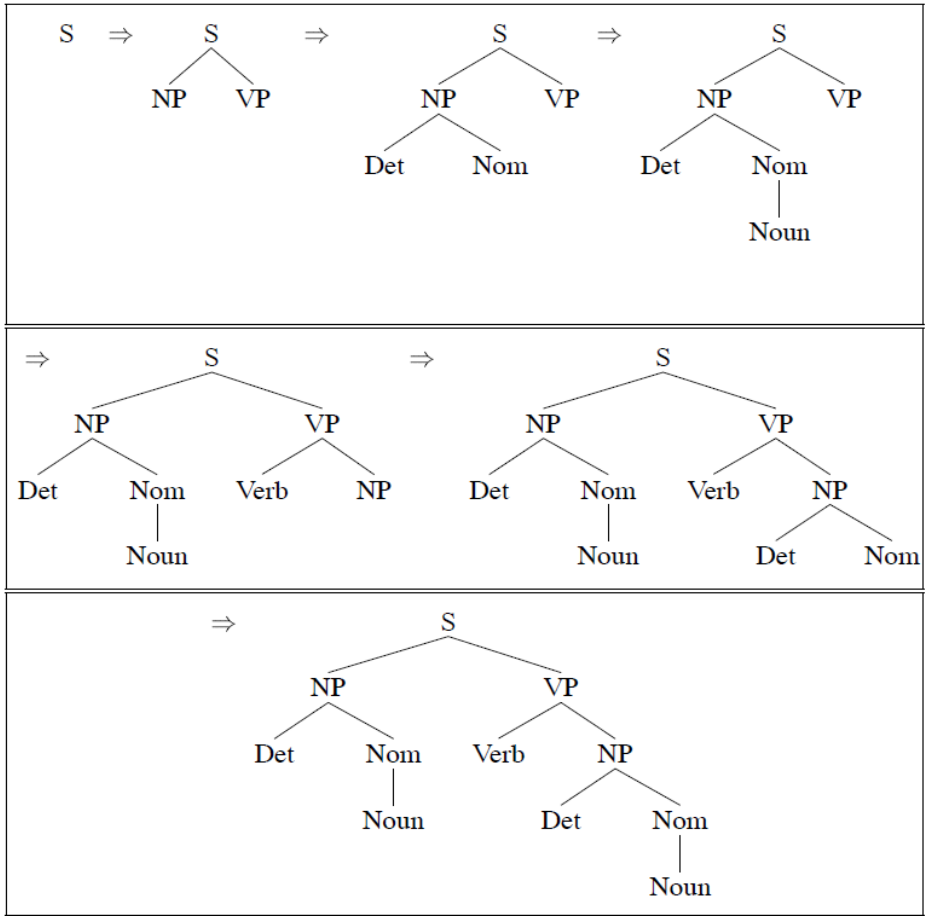
تجزیه به عنوان یک روش جستجو . . .

- جستجوی کلیه حالت‌ها (Naive)
  - نیاز به حافظه و پردازش زیاد
    - به ویژه در گرامر واقعی که درخت‌های مختلف (با دلیل ابهام زیاد) تولید می‌شود.

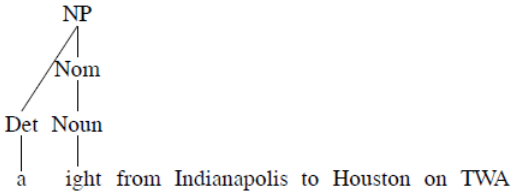
- استفاده از استراتژی‌های جستجو (Backtracking)
  - جستجوی اول عمق (DFS: Depth-First Search)
    - از یک سمت (چپ) یک شاخه انتخاب می‌شود و با حرکت در عمق آنقدر گسترش می‌یابد تا به دنباله کلمات (تایید) یا به بن‌بست (رد) برسد. در صورت رد، شاخه بعدی انتخاب می‌شود.
    - پیاده‌سازی با پشته (stack)
  - جستجوی اول عرض (BFS: Breadth-First Search)
    - همه شاخه‌ها به ترتیب انتخاب می‌شوند و به موازات هم (در عرض) گسترش می‌یابند تا زمانی که یکی از آنها به موفقیت برسد.
    - پیاده‌سازی با صف (queue)

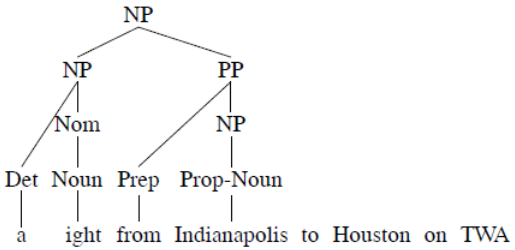
تجزیه به عنوان یک روش جستجو

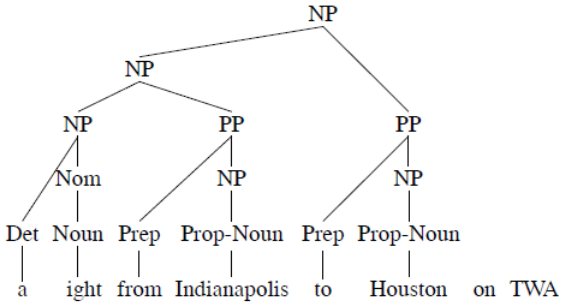
- جستجوی اول عمق (DFS) بالا به پایین

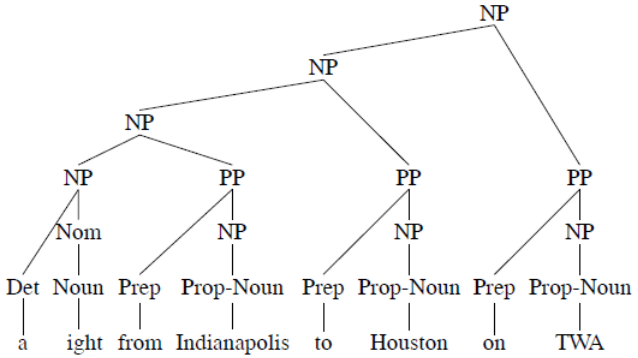


تجزیه به عنوان یک روش جستجو









- جستجوی اول عمق (DFS) بالا به پایین
  - مشکل: تکرار در تولید مجدد زیردرخت‌ها
- نیاز به روشی عمومی و بهینه برای تجزیه
  - ذخیره زیردرخت‌ها بعد از ایجاد در بار اول
    - و عدم تولید مجدد آنها
  - روش مبتنی بر برنامه نویسی پویا (DP)
    - مشابه ویتربی و MED

تجزیه نحوی: رویکردها . . .

- جستجوی ساده (Naive)
  - در نظر گرفتن همه حالت‌ها
- استفاده از استراتژی‌های جستجو (Backtracking)
  - در نظر گرفتن برخی از راه‌حل‌ها و صرفنظر کردن از راه‌حل‌ها جزئی
- مبتنی بر برنامه‌نویسی پویا
  - ذخیره کردن و در نظر گرفتن راه‌حل‌ها جزئی
  - CKY (Cocke-Kasami-Younger)
    - روش پایین به بالا، در سال ۱۹۶۵
  - Earley
    - روش بالا به پایین، توسط Earley در ۱۹۷۰
  - Chart
    - توسط Kay و همکاران، ۱۹۸۶

تجزیه CKY ...

- روش CKY (Cocke-Kasami-Younger)
  - روش پایین به بالا- مبتنی بر برنامه‌نویسی پویا
    - تولید پاسخ با استفاده از زیرپاسخ‌ها
      - یادآوری DP: کاهش زمان الگوریتم‌ها با شکستن یک مساله به زیرمساله‌های آن

- ایده CKY: شکستن دنباله رشته $[s_i, ..., s_k, ..., s_j]$ به زیررشته‌های $[s_i, ..., s_k]$ و $[s_k, ..., s_j]$ و تجزیه (parse) زیررشته‌ها جهت ساخت تجزیه رشته اصلی
  - اگر قانون گرامری مانند $A \rightarrow B\ C$ برای دنباله [i,j] اعمال شود، آنگاه kای وجود دارد که B معادل [i,k] و C معادل [k,j] است
    - $i<k<j$
    - از CNF داریم $j>i+1$

- نیاز است گرامر به صورت نرمال چامسکی (CNF) باشد
  - باید ابتدا گرامر CFG را به CNF تبدیل کرد
  - امکان توسعه برای پشتیبانی از گرامرهای CFG

تجزیه CKY: مثال . . .

- گرامر
  - تبدیل شده به CNF

$S \rightarrow \varepsilon \mid AB \mid XB$

$T \rightarrow AB \mid XB$

$X \rightarrow AT$

$A \rightarrow a$

$B \rightarrow b$

- سوال ۱: آیا رشته w= aaabb توسط گرامر پذیرفته می‌شود؟
- سوال ۲: آیا رشته w= aaabbb توسط گرامر پذیرفته می‌شود؟

تجزیه CKY: مثال . . .

- سوال ۱: آیا رشته w= aaabb توسط گرامر پذیرفته می‌شود؟

❶ الگوریتم پایین به بالاست: شروع از پایین درخت (کلمات/دنباله ورودی)

a	a	a	b	b

$S \rightarrow \varepsilon \mid AB \mid XB$

$T \rightarrow AB \mid XB$

$X \rightarrow AT$

$A \rightarrow a$

$B \rightarrow b$

تجزیه CKY: مثال . . .

- سوال ۱: آیا رشته w= aaabb توسط گرامر پذیرفته می‌شود؟

❷ نوشتن نمادها (متغیرها) برای کلیه رشته‌های با طول ۱

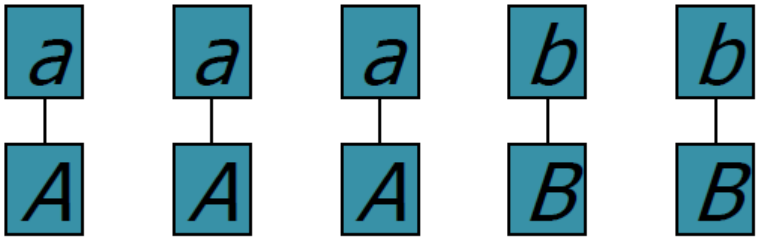
$S \rightarrow \varepsilon \mid AB \mid XB$

$T \rightarrow AB \mid XB$

$X \rightarrow AT$

$A \rightarrow a$

$B \rightarrow b$



تجزیه CKY: مثال . . .

- سوال ۱: آیا رشته w= aaabb توسط گرامر پذیرفته می‌شود؟

❸ نوشتن نمادها (متغیرها) برای کلیه رشته‌های با طول ۲

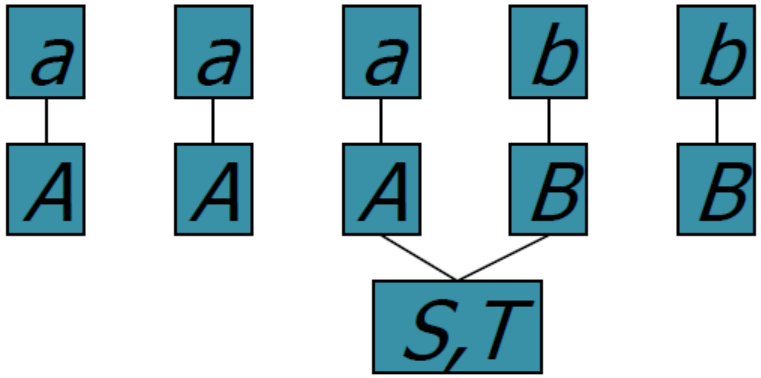
$S \rightarrow \varepsilon \mid AB \mid XB$

$T \rightarrow AB \mid XB$

$X \rightarrow AT$

$A \rightarrow a$

$B \rightarrow b$



تجزیه CKY: مثال . . .

- سوال ۱: آیا رشته w= aaabb توسط گرامر پذیرفته می‌شود؟

❹ نوشتن نمادها (متغیرها) برای کلیه رشته‌های با طول ۳

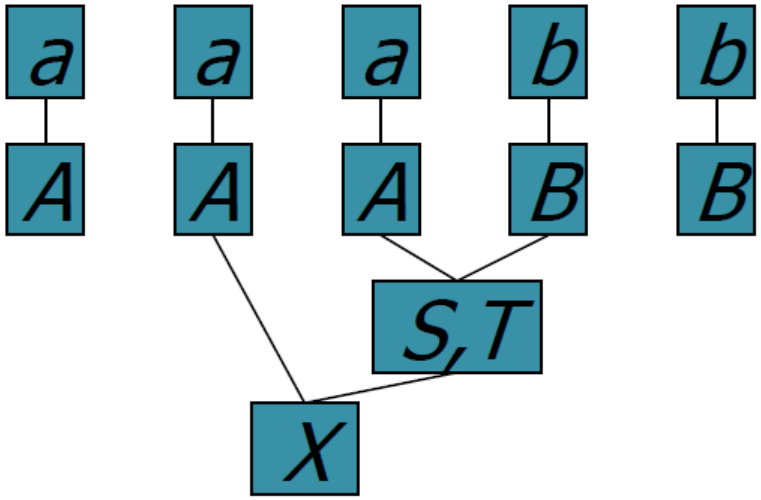
$S \rightarrow \varepsilon \mid AB \mid XB$

$T \rightarrow AB \mid XB$

$X \rightarrow AT$

$A \rightarrow a$

$B \rightarrow b$



تجزیه CKY: مثال . . .

- سوال ۱: آیا رشته w= aaabb توسط گرامر پذیرفته می‌شود؟

❺ نوشتن نمادها (متغیرها) برای کلیه رشته‌های با طول ۴

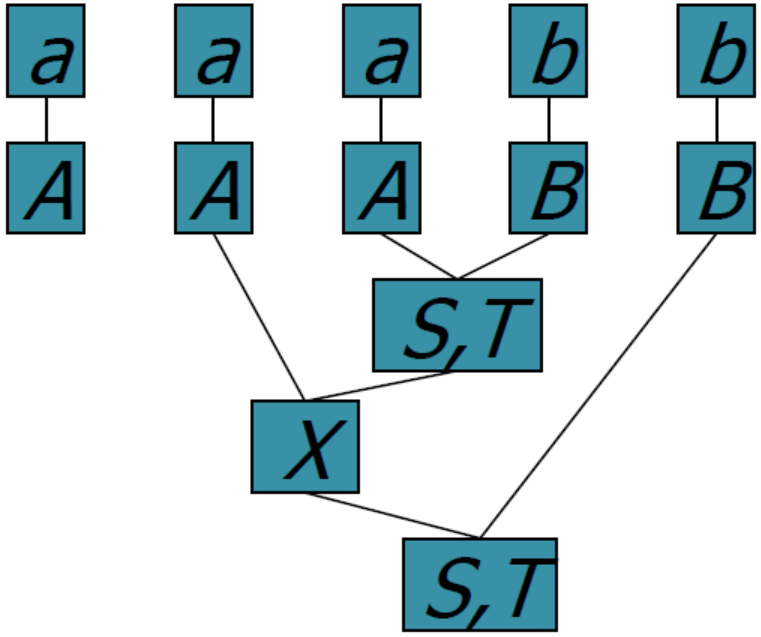
$S \rightarrow \varepsilon \mid AB \mid XB$

$T \rightarrow AB \mid XB$

$X \rightarrow AT$

$A \rightarrow a$

$B \rightarrow b$



تجزیه CKY: مثال . . .

- سوال ۱: آیا رشته w= aaabb توسط گرامر پذیرفته می‌شود؟

❻ نوشتن نمادها (متغیرها) برای کلیه رشته‌های با طول ۵

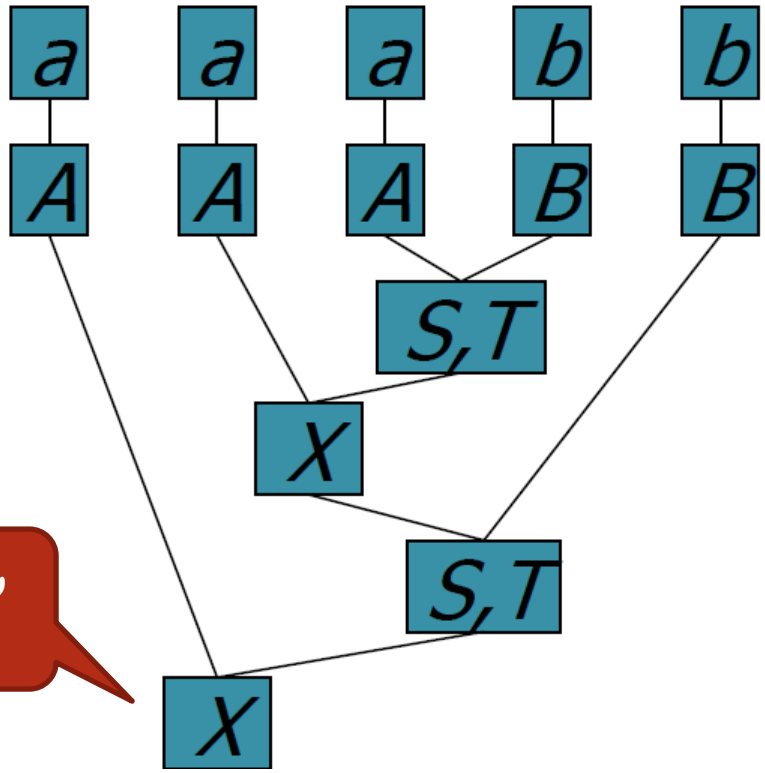
$S \rightarrow \varepsilon \mid AB \mid XB$

$T \rightarrow AB \mid XB$

$X \rightarrow AT$

$A \rightarrow a$

$B \rightarrow b$



رشته w= aaabb پذیرفته نشد چون به S نرسیده‌ایم

تجزیه CKY: مثال . . .

- سوال ۲: آیا رشته w= aaabbb توسط گرامر پذیرفته می‌شود؟

❶ الگوریتم پایین به بالاست: شروع از پایین درخت (کلمات/دنباله ورودی)

$S \rightarrow \varepsilon \mid AB \mid XB$

$T \rightarrow AB \mid XB$

$X \rightarrow AT$

$A \rightarrow a$

$B \rightarrow b$

a	a	a	b	b	b

تجزیه CKY: مثال . . .

- سوال ۲: آیا رشته w= aaabbb توسط گرامر پذیرفته می‌شود؟

❷ نوشتن نمادها (متغیرها) برای کلیه رشته‌های با طول ۱

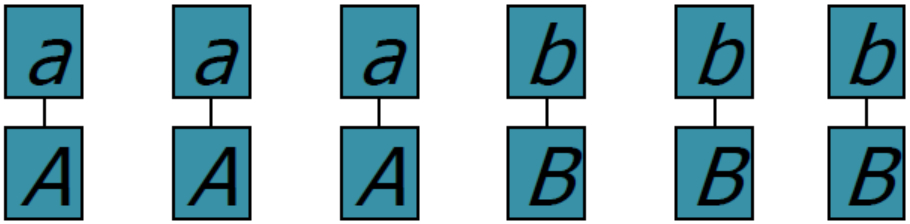
$S \rightarrow \varepsilon \mid AB \mid XB$

$T \rightarrow AB \mid XB$

$X \rightarrow AT$

$A \rightarrow a$

$B \rightarrow b$



تجزیه CKY: مثال . . .

- سوال ۲: آیا رشته w= aaabbb توسط گرامر پذیرفته می‌شود؟

❸ نوشتن نمادها (متغیرها) برای کلیه رشته‌های با طول ۲

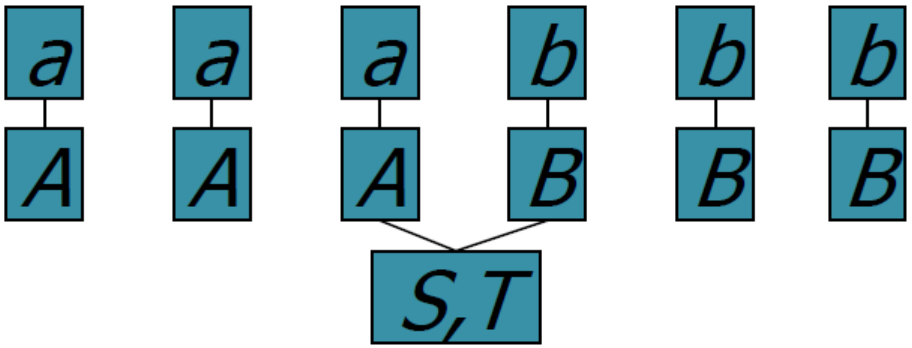
$S \rightarrow \varepsilon \mid AB \mid XB$

$T \rightarrow AB \mid XB$

$X \rightarrow AT$

$A \rightarrow a$

$B \rightarrow b$



تجزیه CKY: مثال . . .

- سوال ۲: آیا رشته w= aaabbb توسط گرامر پذیرفته می‌شود؟

❹ نوشتن نمادها (متغیرها) برای کلیه رشته‌های با طول ۳

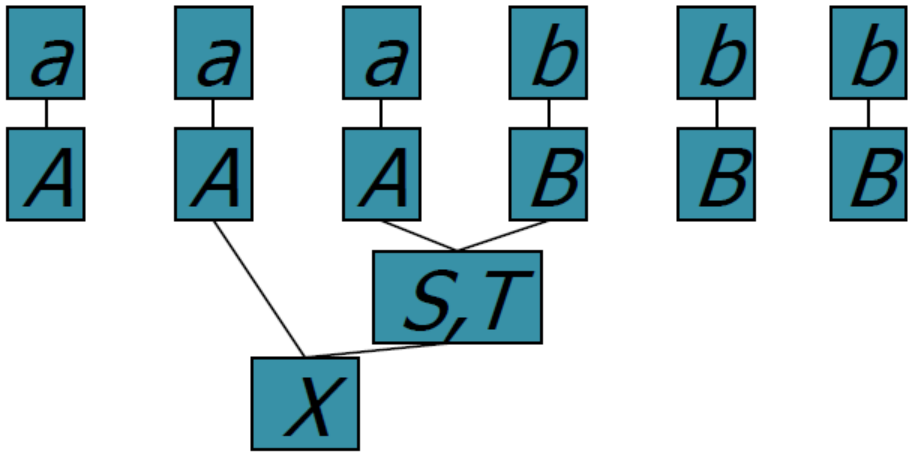
$S \rightarrow \varepsilon \mid AB \mid XB$

$T \rightarrow AB \mid XB$

$X \rightarrow AT$

$A \rightarrow a$

$B \rightarrow b$



تجزیه CKY: مثال . . .

- سوال ۲: آیا رشته w= aaabbb توسط گرامر پذیرفته می‌شود؟

❺ نوشتن نمادها (متغیرها) برای کلیه رشته‌های با طول ۴

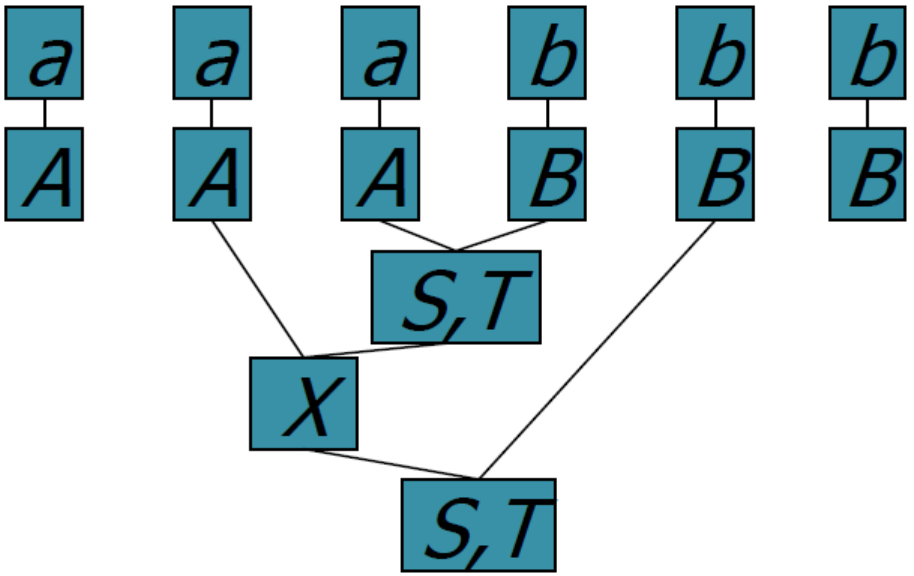
$S \rightarrow \varepsilon \mid AB \mid XB$

$T \rightarrow AB \mid XB$

$X \rightarrow AT$

$A \rightarrow a$

$B \rightarrow b$



تجزیه CKY: مثال . . .

- سوال ۲: آیا رشته w= aaabbb توسط گرامر پذیرفته می‌شود؟

❻ نوشتن نمادها (متغیرها) برای کلیه رشته‌های با طول ۵

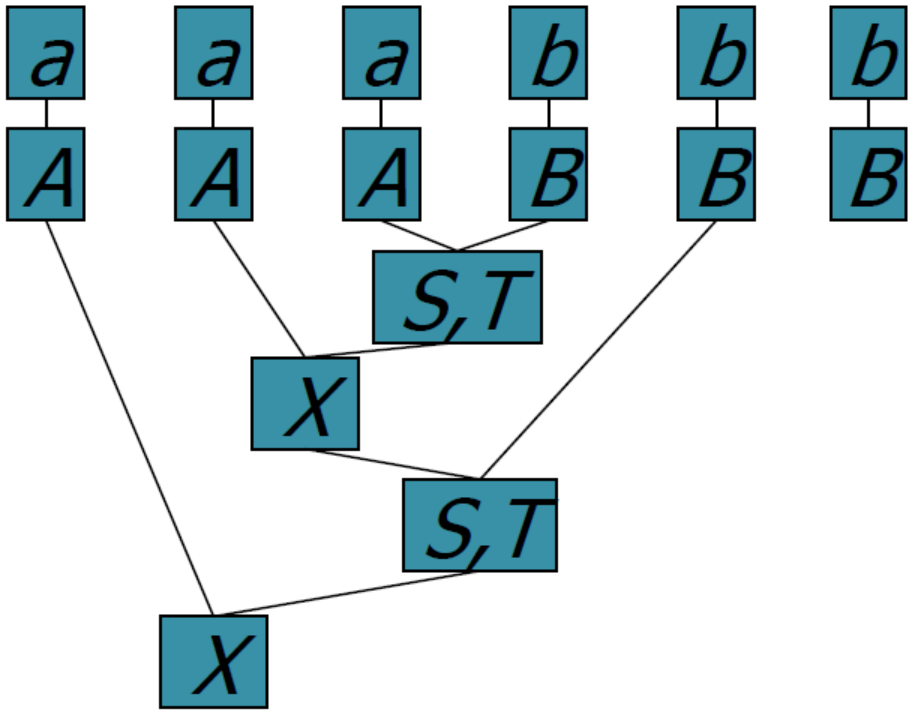
$S \rightarrow \varepsilon \mid AB \mid XB$

$T \rightarrow AB \mid XB$

$X \rightarrow AT$

$A \rightarrow a$

$B \rightarrow b$



تجزیه CKY: مثال . . .

- سوال ۲: آیا رشته w= aaabbb توسط گرامر پذیرفته می‌شود؟

❼ نوشتن نمادها (متغیرها) برای کلیه رشته‌های با طول ۶

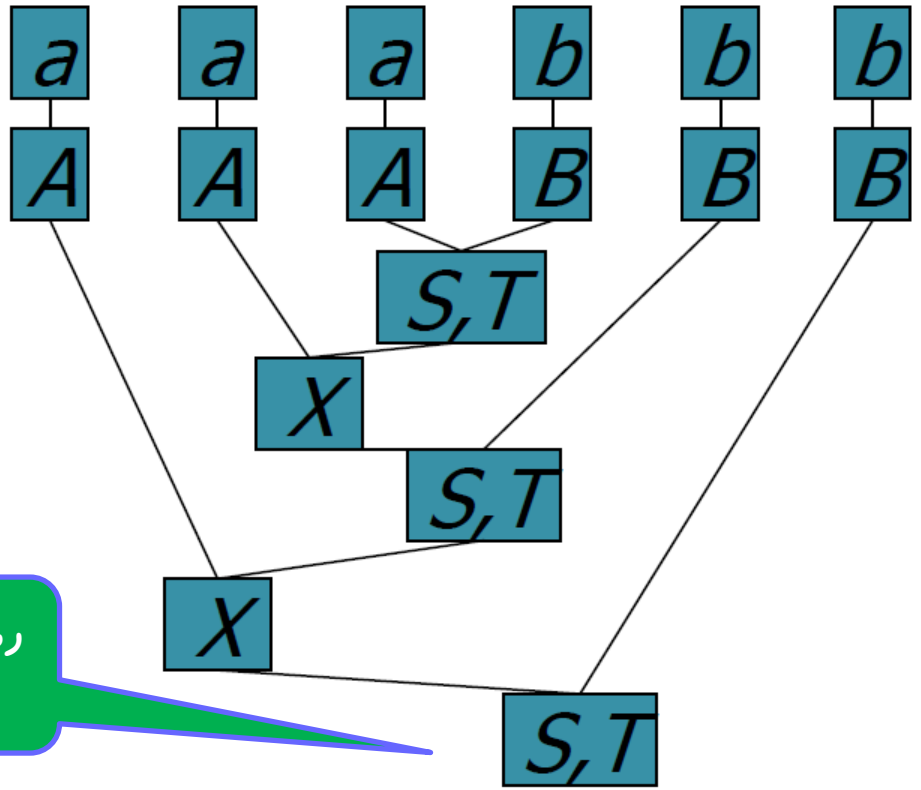
$S \rightarrow \varepsilon \mid AB \mid XB$

$T \rightarrow AB \mid XB$

$X \rightarrow AT$

$A \rightarrow a$

$B \rightarrow b$



رشته w= aaabbb پذیرفته شد چون به S رسیده‌ایم

تجزیه CKY ...

- برای پیاده‌سازی
  - با فرض اینکه طول دنباله ورودی (تعداد کلمات) n باشد
  - یک ماتریس (n+1)*(n+1) ساخته می‌شود (table)
    - اندیس‌ها از 0 تا n
  - اندیس‌ها معادل موقعیت بین کلمات است
    - 0 Book 1 That 2 Flight 3

- خانه [i,j] در ماتریس حاوی غیرپایانه‌های کلیه اجزای پیمایش کننده دنباله ورودی [i,j] است
  - خانه [j-1,j] حاوی پیش پایانه ها است

- اگر خانه [0,n] حاوی غیرپایانه S باشد، جمله پذیرش شده است

تجزیه CKY ...

- الگوریتم پذیرش/تشخیص (Recognition)

function CKY-Parse (*words*, *grammar*) **returns** *table*

for j ← 1 **to** length(*words*) **do**: (loop over columns)

table[j-1,j] ← {$A|A \rightarrow words[j] \in grammar$} (add POS)

for i ← j-2 **downto** 0 **do**: (loop over rows, backwards)

for k ← i+1 to j-1 **do**: (loop over contents of cell)

table[i,j] ← *table*[i,j] ∪

{$A|A \rightarrow B \;\; C \in grammar$,

B ∈ *table*[i,k]

C ∈ *table*[k,j] }

پر کردن قطر اصلی: افزودن همه برچسب‌های ممکن کلمه

تجزیه CKY . . .

- مثال . . .
  - جمله: the chef eats fish with the chopsticks
    - 0 the 1 chef 2 eats 3 fish 4 with 5 the 6 chopsticks 7
  - گرامر

$S \rightarrow NP\ VBZ$
$S \rightarrow NP\ VP$
$VP \rightarrow VP\ PP$
$VP \rightarrow VBZ\ NP$
$VP \rightarrow VBZ\ PP$
$VP \rightarrow VBZ\ NNS$
$VP \rightarrow VBZ\ VP$
$VP \rightarrow VBP\ NP$
$VP \rightarrow VBP\ PP$
$NP \rightarrow DT\ NN$
$NP \rightarrow DT\ NNS$
$PP \rightarrow IN\ NP$

$DT \rightarrow the$
$NN \rightarrow chef$
$NNS \rightarrow fish$
$NNS \rightarrow chopsticks$
$VBP \rightarrow fish$
$VBZ \rightarrow eats$
$IN \rightarrow with$

تجزیه CKY: مثال . . .

- ساخت جدول
  - باید 8*8 باشد (چون تعداد کلمات ورودی ۷ است)

	0	1 the	2 chef	3 eats	4 fish	5 with	6 the	7 chopsticks
0		[0,1]						
1			[1,2]					
2				[2,3]				
3					[3,4]			
4						[4,5]		
5							[5,6]	
6								[6,7]
7								

تجزیه CKY: مثال . . .

```
function CKY-Parse (words, grammar) returns table

  for j ← 1 to length(words) do: (loop over columns)
    table[j-1,j] ← {A|A → words[j] ∈ grammar} (add POS)
    for i ← j-2 downto 0 do: (loop over rows, backwards)
      for k ← i+1 to j-1 do: (loop over contents of cell)
        table[i,j] ← table[i,j] ∪
                {A|A → B  C ∈ grammar,
                B ∈ table[i,k]
                C ∈ table[k,j] }
```

S → NP VBZ	DT → the
S → NP VP	NN → chef
VP → VP PP	NNS → fish
VP → VBZ NP	NNS → chopsticks
VP → VBZ PP	VBP → fish
VP → VBZ NNS	VBZ → eats
VP → VBZ VP	IN → with
VP → VBP NP	
VP → VBP PP	
NP → DT NN	
NP → DT NNS	
PP → IN NP	

- اجرای الگوریتم
  - کلمه the دارای برچسب DT است
    - J=1، i=-1

	0	1 the	2 chef	3 eats	4 fish	5 with	6 the	7 chopsticks
0		DT [0,1]						
1			[1,2]					
2				[2,3]				
3					[3,4]			
4						[4,5]		
5							[5,6]	
6								[6,7]
7								

تجزیه CKY: مثال . . .

```
function CKY-Parse (words, grammar) returns table

  for j ← 1 to length(words) do: (loop over columns)
    table[j-1,j] ← {A|A → words[j] ∈ grammar} (add POS)
    for i ← j-2 downto 0 do: (loop over rows, backwards)
      for k ← i+1 to j-1 do: (loop over contents of cell)
        table[i,j] ← table[i,j] ∪
                {A|A → B  C ∈ grammar,
                B ∈ table[i,k]
                C ∈ table[k,j] }
```

S → NP VBZ	DT → the
S → NP VP	NN → chef
VP → VP PP	NNS → fish
VP → VBZ NP	NNS → chopsticks
VP → VBZ PP	VBP → fish
VP → VBZ NNS	VBZ → eats
VP → VBZ VP	IN → with
VP → VBP NP	
VP → VBP PP	
NP → DT NN	
NP → DT NNS	
PP → IN NP	

- اجرای الگوریتم
  - کلمه chef دارای برچسپ NN است
    - J=2

	0	1 the	2 chef	3 eats	4 fish	5 with	6 the	7 chopsticks
0		DT [0,1]						
1			NN [1,2]					
2				[2,3]				
3					[3,4]			
4						[4,5]		
5							[5,6]	
6								[6,7]
7								

تجزیه CKY: مثال . . .

```
function CKY-Parse (words, grammar) returns table

    for j ← 1 to length(words) do: (loop over columns)

        table[j-1,j] ← {A|A → words[j] ∈ grammar} (add POS)

        for i ← j-2 downto 0 do: (loop over rows, backwards)

            for k ← i+1 to j-1 do: (loop over contents of cell)

                table[i,j] ← table[i,j] ∪

                        {A|A → B  C ∈ grammar,

                        B ∈ table[i,k]

                        C ∈ table[k,j] }
```

S → NP VBZ	DT → the
S → NP VP	NN → chef
VP → VP PP	NNS → fish
VP → VBZ NP	NNS → chopsticks
VP → VBZ PP	VBP → fish
VP → VBZ NNS	VBZ → eats
VP → VBZ VP	IN → with
VP → VBP NP	
VP → VBP PP	
NP → DT NN	
NP → DT NNS	
PP → IN NP	

- اجرای الگوریتم
  - بررسی chef با کلمات قبلی
    - J=2، i=0 و k=1

	0	1 the	2 chef	3 eats	4 fish	5 with	6 the	7 chopsticks
0		DT [0,1] →	NP [0,2]					
1			NN [1,2]					
2				[2,3]				
3					[3,4]			
4						[4,5]		
5							[5,6]	
6								[6,7]
7								

وجود NP→DT NN در گرامر
[0,2]: [0,1], [1,2]

تجزیه CKY: مثال . . .

```
function CKY-Parse (words, grammar) returns table

  for j ← 1 to length(words) do: (loop over columns)
    table[j-1,j] ← {A|A → words[j] ∈ grammar} (add POS)
    for i ← j-2 downto 0 do: (loop over rows, backwards)
      for k ← i+1 to j-1 do: (loop over contents of cell)
        table[i,j] ← table[i,j] ∪
                {A|A → B  C ∈ grammar,
                B ∈ table[i,k]
                C ∈ table[k,j] }
```

S → NP VBZ	DT → the	
S → NP VP	NN → chef	
VP → VP PP	NNS → fish	
VP → VBZ NP	NNS → chopsticks	
VP → VBZ PP	VBP → fish	
VP → VBZ NNS	VBZ → eats	
VP → VBZ VP	IN → with	
VP → VBP NP		
VP → VBP PP		
NP → DT NN		
NP → DT NNS		
PP → IN NP		

- اجرای الگوریتم
  - کلمه eats دارای برچسپ VBZ است
    - J=3

	0	1 the	2 chef	3 eats	4 fish	5 with	6 the	7 chopsticks
0		DT [0,1]	NP [0,2]					
1			NN [1,2]					
2				VBZ [2,3]				
3					[3,4]			
4						[4,5]		
5							[5,6]	
6								[6,7]
7								

تجزیه CKY: مثال . . .

```
function CKY-Parse (words, grammar) returns table

    for j ← 1 to length(words) do: (loop over columns)

        table[j-1,j] ← {A|A → words[j] ∈ grammar} (add POS)

        for i ← j-2 downto 0 do: (loop over rows, backwards)

            for k ← i+1 to j-1 do: (loop over contents of cell)

                table[i,j] ← table[i,j] ∪

                        {A|A → B  C ∈ grammar,

                        B ∈ table[i,k]

                        C ∈ table[k,j] }
```

S → NP VBZ	DT → the
S → NP VP	NN → chef
VP → VP PP	NNS → fish
VP → VBZ NP	NNS → chopsticks
VP → VBZ PP	VBP → fish
VP → VBZ NNS	VBZ → eats
VP → VBZ VP	IN → with
VP → VBP NP	
VP → VBP PP	
NP → DT NN	
NP → DT NNS	
PP → IN NP	

- اجرای الگوریتم
  - بررسی eats با کلمات قبلی
    - J=3، i=1:0 و k=2:2

	0	1 the	2 chef	3 eats	4 fish	5 with	6 the	7 chopsticks
0		DT [0,1]	NP [0,2] →	S [0,3]				
1			NN [1,2]					
2				VBZ [2,3]				
3					[3,4]			
4						[4,5]		
5							[5,6]	
6								[6,7]
7								

وجود S→NP VBZ در گرامر
[0,3]: [0,2], [2,3]

تجزیه CKY: مثال . . .

```
function CKY-Parse (words, grammar) returns table

  for j ← 1 to length(words) do: (loop over columns)
    table[j-1,j] ← {A|A → words[j] ∈ grammar} (add POS)
    for i ← j-2 downto 0 do: (loop over rows, backwards)
      for k ← i+1 to j-1 do: (loop over contents of cell)
        table[i,j] ← table[i,j] ∪
                {A|A → B  C ∈ grammar,
                B ∈ table[i,k]
                C ∈ table[k,j] }
```

S → NP VBZ	DT → the	
S → NP VP	NN → chef	
VP → VP PP	NNS → fish	
VP → VBZ NP	NNS → chopsticks	
VP → VBZ PP	VBP → fish	
VP → VBZ NNS	VBZ → eats	
VP → VBZ VP	IN → with	
VP → VBP NP		
VP → VBP PP		
NP → DT NN		
NP → DT NNS		
PP → IN NP		

- اجرای الگوریتم
  - کلمه fish دارای برچسپ NNS است
    - J=4

	0	1 the	2 chef	3 eats	4 fish	5 with	6 the	7 chopsticks
0		DT [0,1]	NP [0,2]	S [0,3]				
1			NN [1,2]					
2				VBZ [2,3]				
3					NNS [3,4]			
4						[4,5]		
5							[5,6]	
6								[6,7]
7								

تجزیه CKY: مثال ...

```
function CKY-Parse (words, grammar) returns table

  for j ← 1 to length(words) do: (loop over columns)
    table[j-1,j] ← {A|A → words[j] ∈ grammar} (add POS)
    for i ← j-2 downto 0 do: (loop over rows, backwards)
      for k ← i+1 to j-1 do: (loop over contents of cell)
        table[i,j] ← table[i,j] ∪
                {A|A → B  C ∈ grammar,
                B ∈ table[i,k]
                C ∈ table[k,j] }
```

S → NP VBZ	DT → the	
S → NP VP	NN → chef	
VP → VP PP	NNS → fish	
VP → VBZ NP	NNS → chopsticks	
VP → VBZ PP	VBP → fish	
VP → VBZ NNS	VBZ → eats	
VP → VBZ VP	IN → with	
VP → VBP NP		
VP → VBP PP		
NP → DT NN		
NP → DT NNS		
PP → IN NP		

- اجرای الگوریتم
  - کلمه fish دارای برچسپ VBP است
    - J=4

	0	1 the	2 chef	3 eats	4 fish	5 with	6 the	7 chopsticks
0		DT [0,1]	NP [0,2]	S [0,3]				
1			NN [1,2]					
2				VBZ [2,3]				
3					NNS, VBP [3,4]			
4						[4,5]		
5							[5,6]	
6								[6,7]
7								

تجزیه CKY: مثال . . .

```
function CKY-Parse (words, grammar) returns table

    for j ← 1 to length(words) do: (loop over columns)

        table[j-1,j] ← {A|A → words[j] ∈ grammar} (add POS)

        for i ← j-2 downto 0 do: (loop over rows, backwards)

            for k ← i+1 to j-1 do: (loop over contents of cell)

                table[i,j] ← table[i,j] ∪

                        {A|A → B  C ∈ grammar,

                        B ∈ table[i,k]

                        C ∈ table[k,j] }
```

S → NP VBZ	DT → the
S → NP VP	NN → chef
VP → VP PP	NNS → fish
VP → VBZ NP	NNS → chopsticks
VP → VBZ PP	VBP → fish
VP → VBZ NNS	VBZ → eats
VP → VBZ VP	IN → with
VP → VBP NP	
VP → VBP PP	
NP → DT NN	
NP → DT NNS	
PP → IN NP	

- اجرای الگوریتم
  - بررسی fish با کلمات قبلی
    - J=4، i=2:0 و k=3

	0	1 the	2 chef	3 eats	4 fish	5 with	6 the	7 chopsticks
0		DT [0,1]	NP [0,2]	S [0,3]				
1			NN [1,2]					
2				VBZ [2,3]	VP [2,4]			
3					NNS, VBP [3,4]			
4						[4,5]		
5							[5,6]	
6								[6,7]
7								

وجود VP→VBZ NNS در گرامر
[2,4]: [2,3], [3,4]

تجزیه CKY: مثال . . .

```
function CKY-Parse (words, grammar) returns table
    for j ← 1 to length(words) do: (loop over columns)
        table[j-1,j] ← {A|A → words[j] ∈ grammar} (add POS)
        for i ← j-2 downto 0 do: (loop over rows, backwards)
            for k ← i+1 to j-1 do: (loop over contents of cell)
                table[i,j] ← table[i,j] ∪
                    {A|A → B  C ∈ grammar,
                    B ∈ table[i,k]
                    C ∈ table[k,j] }
```

S → NP VBZ	DT → the
S → NP VP	NN → chef
VP → VP PP	NNS → fish
VP → VBZ NP	NNS → chopsticks
VP → VBZ PP	VBP → fish
VP → VBZ NNS	VBZ → eats
VP → VBZ VP	IN → with
VP → VBP NP	
VP → VBP PP	
NP → DT NN	
NP → DT NNS	
PP → IN NP	

- اجرای الگوریتم
  - بررسی fish با کلمات قبلی
    - J=4، i=2:0 و k=2

	0	1 the	2 chef	3 eats	4 fish	5 with	6 the	7 chopsticks
0		DT [0,1]	NP [0,2]	S [0,3]	S [0,4]			
1			NN [1,2]					
2				VBZ [2,3]	VP [2,4]			
3					NNS, VBP [3,4]			
4						[4,5]		
5							[5,6]	
6								[6,7]
7								

وجود S→NP VP در گرامر
[0,4]: [0,2], [2,4]

تجزیه CKY: مثال . . .

```
function CKY-Parse (words, grammar) returns table

  for j ← 1 to length(words) do: (loop over columns)
    table[j-1,j] ← {A|A → words[j] ∈ grammar} (add POS)
    for i ← j-2 downto 0 do: (loop over rows, backwards)
      for k ← i+1 to j-1 do: (loop over contents of cell)
        table[i,j] ← table[i,j] ∪
                {A|A → B  C ∈ grammar,
                B ∈ table[i,k]
                C ∈ table[k,j] }
```

S → NP VBZ	DT → the
S → NP VP	NN → chef
VP → VP PP	NNS → fish
VP → VBZ NP	NNS → chopsticks
VP → VBZ PP	VBP → fish
VP → VBZ NNS	VBZ → eats
VP → VBZ VP	IN → with
VP → VBP NP	
VP → VBP PP	
NP → DT NN	
NP → DT NNS	
PP → IN NP	

- اجرای الگوریتم
  - کلمه with دارای برچسپ IN است
    - J=5

	0	1 the	2 chef	3 eats	4 fish	5 with	6 the	7 chopsticks
0		DT [0,1]	NP [0,2]	S [0,3]	S [0,4]			
1			NN [1,2]					
2				VBZ [2,3]	VP [2,4]			
3					NNS, VBP [3,4]			
4						IN [4,5]		
5							[5,6]	
6								[6,7]
7								

تجزیه CKY: مثال . . .

```
function CKY-Parse (words, grammar) returns table

    for j ← 1 to length(words) do: (loop over columns)

        table[j-1,j] ← {A|A → words[j] ∈ grammar} (add POS)

        for i ← j-2 downto 0 do: (loop over rows, backwards)

            for k ← i+1 to j-1 do: (loop over contents of cell)

                table[i,j] ← table[i,j] ∪

                        {A|A → B  C ∈ grammar,

                        B ∈ table[i,k]

                        C ∈ table[k,j] }
```

S → NP VBZ	DT → the	
S → NP VP	NN → chef	
VP → VP PP	NNS → fish	
VP → VBZ NP	NNS → chopsticks	
VP → VBZ PP	VBP → fish	
VP → VBZ NNS	VBZ → eats	
VP → VBZ VP	IN → with	
VP → VBP NP		
VP → VBP PP		
NP → DT NN		
NP → DT NNS		
PP → IN NP		

- اجرای الگوریتم
  - بررسی with با کلمات قبلی
    - J=5، i=3:0

	0	1 the	2 chef	3 eats	4 fish	5 with	6 the	7 chopsticks
0		DT [0,1]	NP [0,2]	S [0,3]	S [0,4]			
1			NN [1,2]					
2				VBZ [2,3]	VP [2,4]			
3					NNS, VBP [3,4]			
4						IN [4,5]		
5							[5,6]	
6								[6,7]
7								

عدم تغییر جدول

تجزیه CKY: مثال . . .

```
function CKY-Parse (words, grammar) returns table

  for j ← 1 to length(words) do: (loop over columns)
    table[j-1,j] ← {A|A → words[j] ∈ grammar} (add POS)
    for i ← j-2 downto 0 do: (loop over rows, backwards)
      for k ← i+1 to j-1 do: (loop over contents of cell)
        table[i,j] ← table[i,j] ∪
                {A|A → B  C ∈ grammar,
                B ∈ table[i,k]
                C ∈ table[k,j] }
```

S → NP VBZ	DT → the
S → NP VP	NN → chef
VP → VP PP	NNS → fish
VP → VBZ NP	NNS → chopsticks
VP → VBZ PP	VBP → fish
VP → VBZ NNS	VBZ → eats
VP → VBZ VP	IN → with
VP → VBP NP	
VP → VBP PP	
NP → DT NN	
NP → DT NNS	
PP → IN NP	

- اجرای الگوریتم
  - کلمه the دارای برچسپ DT است
    - J=6

	0	1 the	2 chef	3 eats	4 fish	5 with	6 the	7 chopsticks
0		DT [0,1]	NP [0,2]	S [0,3]	S [0,4]			
1			NN [1,2]					
2				VBZ [2,3]	VP [2,4]			
3					NNS, VBP [3,4]			
4						IN [4,5]		
5							DT [5,6]	
6								[6,7]
7								

تجزیه CKY: مثال . . .

```
function CKY-Parse (words, grammar) returns table

    for j ← 1 to length(words) do: (loop over columns)

        table[j-1,j] ← {A|A → words[j] ∈ grammar} (add POS)

        for i ← j-2 downto 0 do: (loop over rows, backwards)

            for k ← i+1 to j-1 do: (loop over contents of cell)

                table[i,j] ← table[i,j] ∪

                        {A|A → B  C ∈ grammar,

                        B ∈ table[i,k]

                        C ∈ table[k,j] }
```

S → NP VBZ	DT → the
S → NP VP	NN → chef
VP → VP PP	NNS → fish
VP → VBZ NP	NNS → chopsticks
VP → VBZ PP	VBP → fish
VP → VBZ NNS	VBZ → eats
VP → VBZ VP	IN → with
VP → VBP NP	
VP → VBP PP	
NP → DT NN	
NP → DT NNS	
PP → IN NP	

- اجرای الگوریتم
  - بررسی the با کلمات قبلی
    - J=6، i=4:0

	0	1 the	2 chef	3 eats	4 fish	5 with	6 the	7 chopsticks
0		DT [0,1]	NP [0,2]	S [0,3]	S [0,4]			
1			NN [1,2]					
2				VBZ [2,3]	VP [2,4]			
3					NNS, VBP [3,4]			
4						IN [4,5]		
5							DT [5,6]	
6								[6,7]
7								

عدم تغییر جدول

تجزیه CKY: مثال . . .

```
function CKY-Parse (words, grammar) returns table

  for j ← 1 to length(words) do: (loop over columns)
    table[j-1,j] ← {A|A → words[j] ∈ grammar} (add POS)
    for i ← j-2 downto 0 do: (loop over rows, backwards)
      for k ← i+1 to j-1 do: (loop over contents of cell)
        table[i,j] ← table[i,j] ∪
                {A|A → B  C ∈ grammar,
                B ∈ table[i,k]
                C ∈ table[k,j] }
```

S → NP VBZ	DT → the
S → NP VP	NN → chef
VP → VP PP	NNS → fish
VP → VBZ NP	NNS → chopsticks
VP → VBZ PP	VBP → fish
VP → VBZ NNS	VBZ → eats
VP → VBZ VP	IN → with
VP → VBP NP	
VP → VBP PP	
NP → DT NN	
NP → DT NNS	
PP → IN NP	

- اجرای الگوریتم
  - کلمه chopsticks دارای برچسب NNS
  - J=7

	0	1 the	2 chef	3 eats	4 fish	5 with	6 the	7 chopsticks
0		DT [0,1]	NP [0,2]	S [0,3]	S [0,4]			
1			NN [1,2]					
2				VBZ [2,3]	VP [2,4]			
3					NNS, VBP [3,4]			
4						IN [4,5]		
5							DT [5,6]	
6								NNS [6,7]
7								

تجزیه CKY: مثال . . .

```
function CKY-Parse (words, grammar) returns table

    for j ← 1 to length(words) do: (loop over columns)

        table[j-1,j] ← {A|A → words[j] ∈ grammar} (add POS)

        for i ← j-2 downto 0 do: (loop over rows, backwards)

            for k ← i+1 to j-1 do: (loop over contents of cell)

                table[i,j] ← table[i,j] ∪

                        {A|A → B  C ∈ grammar,

                        B ∈ table[i,k]

                        C ∈ table[k,j] }
```

S → NP VBZ	DT → the
S → NP VP	NN → chef
VP → VP PP	NNS → fish
VP → VBZ NP	NNS → chopsticks
VP → VBZ PP	VBP → fish
VP → VBZ NNS	VBZ → eats
VP → VBZ VP	IN → with
VP → VBP NP	
VP → VBP PP	
NP → DT NN	
NP → DT NNS	
PP → IN NP	

- اجرای الگوریتم
  - بررسی chopsticks با کلمات قبلی
    - J=7، i=5:0(5) و k=6

	0	1 the	2 chef	3 eats	4 fish	5 with	6 the	7 chopsticks
0		DT [0,1]	NP [0,2]	S [0,3]	S [0,4]			
1			NN [1,2]					
2				VBZ [2,3]	VP [2,4]			
3					NNS, VBP [3,4]			
4						IN [4,5]		
5							DT [5,6]	NP [5,7]
6								NNS [6,7]
7								

تجزیه CKY: مثال . . .

```
function CKY-Parse (words, grammar) returns table

    for j ← 1 to length(words) do: (loop over columns)

        table[j-1,j] ← {A|A → words[j] ∈ grammar} (add POS)

        for i ← j-2 downto 0 do: (loop over rows, backwards)

            for k ← i+1 to j-1 do: (loop over contents of cell)

                table[i,j] ← table[i,j] ∪

                        {A|A → B  C ∈ grammar,

                        B ∈ table[i,k]

                        C ∈ table[k,j] }
```

S → NP VBZ	DT → the
S → NP VP	NN → chef
VP → VP PP	NNS → fish
VP → VBZ NP	NNS → chopsticks
VP → VBZ PP	VBP → fish
VP → VBZ NNS	VBZ → eats
VP → VBZ VP	IN → with
VP → VBP NP	
VP → VBP PP	
NP → DT NN	
NP → DT NNS	
PP → IN NP	

- اجرای الگوریتم
  - بررسی chopsticks با کلمات قبلی
    - J=7، i=4 و k=4

	0	1 the	2 chef	3 eats	4 fish	5 with	6 the	7 chopsticks
0		DT [0,1]	NP [0,2]	S [0,3]	S [0,4]			
1			NN [1,2]					
2				VBZ [2,3]	VP [2,4]			
3					NNS, VBP [3,4]			
4						IN [4,5]		PP [4,7]
5							DT [5,6]	NP [5,7]
6								NNS [6,7]
7								

تجزیه CKY: مثال . . .

```
function CKY-Parse (words, grammar) returns table

    for j ← 1 to length(words) do: (loop over columns)

        table[j-1,j] ← {A|A → words[j] ∈ grammar} (add POS)

        for i ← j-2 downto 0 do: (loop over rows, backwards)

            for k ← i+1 to j-1 do: (loop over contents of cell)

                table[i,j] ← table[i,j] ∪

                        {A|A → B  C ∈ grammar,

                        B ∈ table[i,k]

                        C ∈ table[k,j] }
```

S → NP VBZ	DT → the
S → NP VP	NN → chef
VP → VP PP	NNS → fish
VP → VBZ NP	NNS → chopsticks
VP → VBZ PP	VBP → fish
VP → VBZ NNS	VBZ → eats
VP → VBZ VP	IN → with
VP → VBP NP	
VP → VBP PP	
NP → DT NN	
NP → DT NNS	
PP → IN NP	

- اجرای الگوریتم
  - بررسی chopsticks با کلمات قبلی
    - J=7، i=3 و k=4

	0	1 the	2 chef	3 eats	4 fish	5 with	6 the	7 chopsticks
0		DT [0,1]	NP [0,2]	S [0,3]	S [0,4]			
1			NN [1,2]					
2				VBZ [2,3]	VP [2,4]			
3					NNS, VBP [3,4]			VP [3,7]
4						IN [4,5]		PP [4,7]
5							DT [5,6]	NP [5,7]
6								NNS [6,7]
7								

تجزیه CKY: مثال . . .

```
function CKY-Parse (words, grammar) returns table

    for j ← 1 to length(words) do: (loop over columns)

        table[j-1,j] ← {A|A → words[j] ∈ grammar} (add POS)

        for i ← j-2 downto 0 do: (loop over rows, backwards)

            for k ← i+1 to j-1 do: (loop over contents of cell)

                table[i,j] ← table[i,j] ∪

                        {A|A → B  C ∈ grammar,

                        B ∈ table[i,k]

                        C ∈ table[k,j] }
```

S → NP VBZ	DT → the
S → NP VP	NN → chef
VP → VP PP	NNS → fish
VP → VBZ NP	NNS → chopsticks
VP → VBZ PP	VBP → fish
VP → VBZ NNS	VBZ → eats
VP → VBZ VP	IN → with
VP → VBP NP	
VP → VBP PP	
NP → DT NN	
NP → DT NNS	
PP → IN NP	

- اجرای الگوریتم
  - بررسی chopsticks با کلمات قبلی
    - J=7، i=2 و k=3

	0	1 the	2 chef	3 eats	4 fish	5 with	6 the	7 chopsticks
0		DT [0,1]	NP [0,2]	S [0,3]	S [0,4]			
1			NN [1,2]					
2				VBZ [2,3]	VP [2,4]			VP [2,7]
3					NNS, VBP [3,4]			VP [3,7]
4						IN [4,5]		PP [4,7]
5							DT [5,6]	NP [5,7]
6								NNS [6,7]
7								

تجزیه CKY: مثال . . .

```
function CKY-Parse (words, grammar) returns table

    for j ← 1 to length(words) do: (loop over columns)

        table[j-1,j] ← {A|A → words[j] ∈ grammar} (add POS)

        for i ← j-2 downto 0 do: (loop over rows, backwards)

            for k ← i+1 to j-1 do: (loop over contents of cell)

                table[i,j] ← table[i,j] ∪

                        {A|A → B  C ∈ grammar,

                        B ∈ table[i,k]

                        C ∈ table[k,j] }
```

S → NP VBZ	DT → the
S → NP VP	NN → chef
VP → VP PP	NNS → fish
VP → VBZ NP	NNS → chopsticks
VP → VBZ PP	VBP → fish
VP → VBZ NNS	VBZ → eats
VP → VBZ VP	IN → with
VP → VBP NP	
VP → VBP PP	
NP → DT NN	
NP → DT NNS	
PP → IN NP	

- اجرای الگوریتم
  - بررسی chopsticks با کلمات قبلی
    - J=7، i=2 و k=4

	0	1 the	2 chef	3 eats	4 fish	5 with	6 the	7 chopsticks
0		DT [0,1]	NP [0,2]	S [0,3]	S [0,4]			
1			NN [1,2]					
2				VBZ [2,3]	VP [2,4]			VP1, VP2 [2,7]
3					NNS, VBP [3,4]			VP [3,7]
4						IN [4,5]		PP [4,7]
5							DT [5,6]	NP [5,7]
6								NNS [6,7]
7								

تجزیه CKY: مثال ...

```
function CKY-Parse (words, grammar) returns table

    for j ← 1 to length(words) do: (loop over columns)

        table[j-1,j] ← {A|A → words[j] ∈ grammar} (add POS)

        for i ← j-2 downto 0 do: (loop over rows, backwards)

            for k ← i+1 to j-1 do: (loop over contents of cell)

                table[i,j] ← table[i,j] ∪

                        {A|A → B  C ∈ grammar,

                        B ∈ table[i,k]

                        C ∈ table[k,j] }
```

S → NP VBZ	DT → the
S → NP VP	NN → chef
VP → VP PP	NNS → fish
VP → VBZ NP	NNS → chopsticks
VP → VBZ PP	VBP → fish
VP → VBZ NNS	VBZ → eats
VP → VBZ VP	IN → with
VP → VBP NP	
VP → VBP PP	
NP → DT NN	
NP → DT NNS	
PP → IN NP	

- اجرای الگوریتم
  - بررسی chopsticks با کلمات قبلی
    - J=7، i=0 و k=2

	0	1 the	2 chef	3 eats	4 fish	5 with	6 the	7 chopsticks
0		DT [0,1]	NP [0,2]	S [0,3]	S [0,4]			S [0,7]
1			NN [1,2]					
2				VBZ [2,3]	VP [2,4]			VP1, VP2 [2,7]
3					NNS, VBP [3,4]			VP [3,7]
4						IN [4,5]		PP [4,7]
5							DT [5,6]	NP [5,7]
6								NNS [6,7]
7								

تجزیه CKY: مثال ...

```
function CKY-Parse (words, grammar) returns table

  for j ← 1 to length(words) do: (loop over columns)

    table[j-1,j] ← {A|A → words[j] ∈ grammar} (add POS)

    for i ← j-2 downto 0 do: (loop over rows, backwards)

      for k ← i+1 to j-1 do: (loop over contents of cell)

        table[i,j] ← table[i,j] ∪

                {A|A → B  C ∈ grammar,

                B ∈ table[i,k]

                C ∈ table[k,j] }
```

S → NP VBZ	DT → the
S → NP VP	NN → chef
VP → VP PP	NNS → fish
VP → VBZ NP	NNS → chopsticks
VP → VBZ PP	VBP → fish
VP → VBZ NNS	VBZ → eats
VP → VBZ VP	IN → with
VP → VBP NP	
VP → VBP PP	
NP → DT NN	
NP → DT NNS	
PP → IN NP	

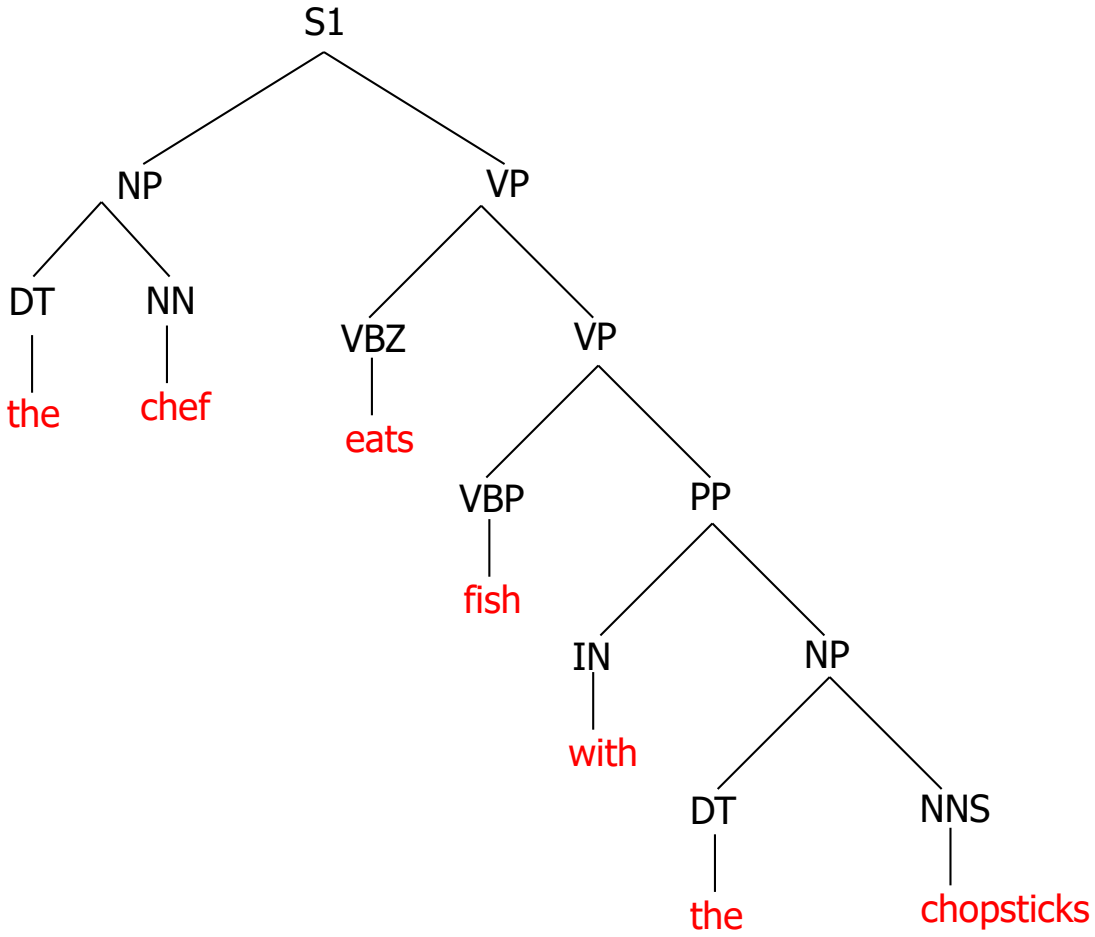
- اجرای الگوریتم
  - بررسی chopsticks با کلمات قبلی
    - J=7، i=0 و k=2

	0	1 the	2 chef	3 eats	4 fish	5 with	6 the	7 chopsticks
0		DT [0,1]	NP [0,2]	S [0,3]	S [0,4]			S1, S2 [0,7]
1			NN [1,2]					
2				VBZ [2,3]	VP [2,4]			VP1, VP2 [2,7]
3					NNS, VBP [3,4]			VP [3,7]
4						IN [4,5]		PP [4,7]
5							DT [5,6]	NP [5,7]
6								NNS [6,7]
7								

پذیرش رشته: در [0,7] علامت S وجود دارد

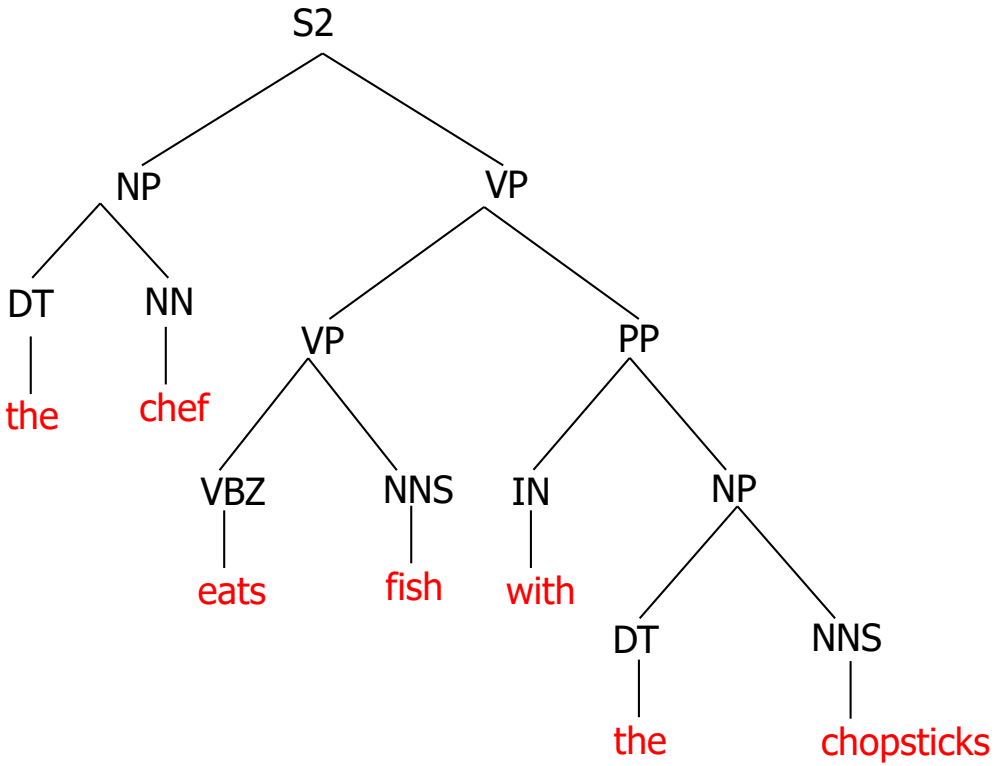
تجزیه CKY: مثال . . .

- درخت تجزیه: ۲ مورد



تجزیه CKY: مثال . . .

- درخت تجزیه: ۲ مورد



تجزیه CKY: مثال . . .

- مفهوم جدول (چارت)

	0	1 the	2 chef	3 eats	4 fish	5 with	6 the	7 chopsticks
0		the	the chef (NP)	the chef eats	the chef eats fish	the chef eats fish with	the chef eats fish with the	the chef eats fish with the chopsticks (S)
1			chef	chef eats	chef eats fish	chef eats fish with	chef eats fish with the	chef eats fish with the chopsticks
2				eats	eats fish	eats fish with	eats fish with the	eats fish with the chopsticks (VP)
3					fish	fish with	fish with the	fish with the chopsticks
4						with	with the	with the chopsticks (PP)
5							the	the chopsticks (NP)
6								chopsticks

نگهداری کلیه اطلاعات میانی (همه زیردرخت ها) و جلوگیری از محاسبه دوباره

تجزیه CKY ...

- اضافه کردن اطلاعات لازم برای نگهداری درخت تجزیه

function CKY-Parse (*words*, *grammar*) **returns** *parses*

for j ← 1 **to** length(*words*) **do**: (loop over columns)

table[j-1,j] ← **for all** $\{A|A \rightarrow words[j] \in grammar\}$ (add all POS)

for i ← j-2 **downto** 0 **do**: (loop over rows, backwards)

for k ← i+1 to j-1 **do**: (loop over contents of cell)

for all $\{A|A \rightarrow B\ C\}$: (all productions)

اطلاعات درخت تجزیه

back[i,j,A] ← { k,B,C } (add back pointer)

return buildtree(*back*[1, length(*words*,S]), *table*[1,LENGTH(*words*),S]
(follow back pointer)

تجزیه CKY

- پیچیدگی زمانی (محاسباتی)
  - تعداد سه حلقه for با حداکثر تکرار n (تعداد کلمات جمله ورودی)
  - در هر مرحله دنبال $r=|N|$ (تعداد کل عناصر غیرپایانه) جفت قانون گرامری است
  - بنابراین پیچیدگی محاسباتی = $O(r^2n^3) = O(n^3)$
- پیچیدگی حافظه
  - یک جدول $(n+1)*(n+1)$ که هر خانه آن حداکثر r عضو را نگه می‌دارد
  - بنابراین پیچیدگی حافظه = $O(rn^2) = O(n^2)$

تجزیه Earley ...

- روش Earley (۱۹۷۰)
  - تجزیه بالا به پایین - مبتنی بر برنامه‌نویسی پویا
- عدم نیاز به صورت CNF
  - صورت گرامری CNF ساختار واقعی گرامری زبان را از بین می‌برد
- برای بسیاری از گرامرها، از $O(n^3)$ سریع‌تر است
  - زیردرخت هایی که پتانسیل پاسخ را ندارند، تولید نمی کند

تجزیه Earley ...

- چارت (Chart)
  - جدولی (N+1)*(N+1) (برای N کلمه)
  - حاوی دنباله‌ای از حالت‌ها که بیانگر درخت تجزیه از ابتدا تا کلمه جاری
  - در پایان دنباله کلمات، Chart حاوی همه تجزیه‌های دنباله کلمات است
  - هر حالت در آرایه چارت حاوی سه نوع اطلاعات است
    - یک زیردرخت متناسب با یک قاعده گرامری
    - اطلاعاتی در مورد نحوه ایجاد زیردرخت جاری
    - موقعیت زیردرخت جاری نسبت به ورودی

1. a subtree corresponding to a grammar rule
   $S \rightarrow NP\ VP$
2. info about progress made towards completing this subtree
   $S \rightarrow NP \bullet VP$
3. the position of the subtree wrt input
   $S \rightarrow NP \bullet VP, [0, 3]$

- قاعده نقطه ای (Dotted Rule)
  - استفاده از یک نقطه • در سمت راست قاعده گرامری مربوط به حالت برای بیان استفاده از آن قاعده
- بیان موقعیت هر حالت با دو عدد
  - یکی از اعداد بیانگر شروع حالت و موقعیت نقطه • است

$$VP \rightarrow V\ NP\bullet, [0, 3]$$

از ابتدا (۰) تا کلمه ۳ یک VP داریم که خود VP از V و NP تشکیل شده است

تجزیه Earley ...

- مفهوم حالت (State)
  - هر عنصر چارت را یک حالت می گویند
  - هر حالت حاوی یکی از سه نوع جزو تشکیل‌دهنده (عبارت) است:

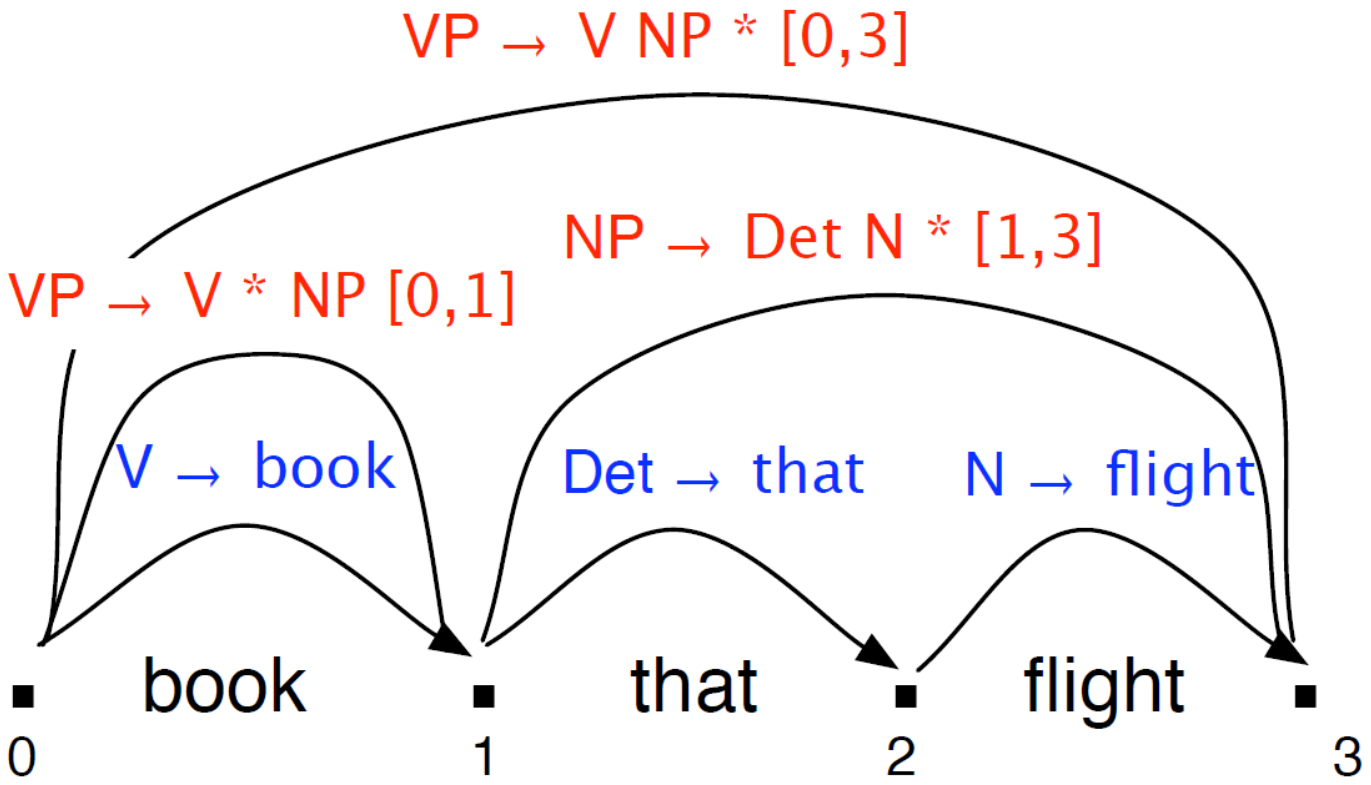
Completed: VP → V NP •
In-progress: NP → Det • Nom
Predicted: S → • VP

  - افزودن مختصات به هر حالت

- $A \rightarrow \alpha,\ [x, y]$
- x indicates the position in input where the state begins
- y indicates position of dot

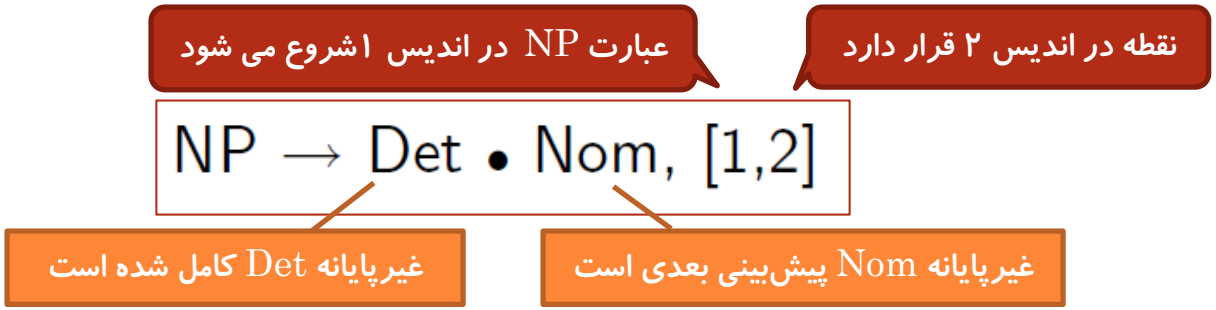
تجزیه Earley ...

- مفهوم حالت (State): مثال



تجزیه Earley ...

- مفهوم حالت (State)
  - برای جمله Book that flight



VP → V NP •, [0,3]

- VP is completed
- no further predictions from this rule
- a successful parse that spans the entire input

تجزیه Earley ...

- موفق شدن تجزیه
  - همه کلمات دیده شده باشد
  - حالت چارت کامل (Completed) باشد (اندیس نقطه N باشد)

  - عبارت زیر (حالت) را در چارت داشته باشیم

$$S \rightarrow \alpha \bullet, [0,N]$$

تجزیه Earley: گام‌های الگوریتم ...

- مقداردهی اولیه
  - یک غیرپایانه مانند X و یک نماد (کلمه) مانند EOS (پایان جمله) اضافه کنیم
  - نماد (کلمه) EOS را به انتهای جمله ورودی اضافه می کنیم
  - همه قوانین گرامری که با ... → S هستند (ریشه درخت)، حالت X → • S EOS را Chart[0,0] اضافه کنیم
- اجرای الگوریتم

Predictor:	Adds predictions into the chart
Completer:	Moves the dot to the right when new constituents are found
Scanner:	Reads the input words and enters states representing those words into the chart

تجزیه Earley: گام‌های الگوریتم . . .

- پیش‌بینی‌گر (Predictor) . . .

```
procedure PREDICTOR((A → α • B β, [i, j]))
    for each (B → γ) in GRAMMAR-RULES-FOR(B, grammar)
do
        ENQUEUE((B → • γ, [j, j]), chart[j])
    end
```

- **برای هر غیرپایانه مانند B که قبل از آن نقطه وجود دارد:**
  - تمام قوانین گرامری که سمت چپ آن B است به حالات چارت اضافه می شود.
  - برای هر حالت جدید، اولین مولفه سمت راست قانون را نقطه قرار دهید.
  - اندیس‌ها تغییر نمی‌کنند.

تجزیه Earley: گام‌های الگوریتم . . .

- پویش‌گر (Scanner) . . .

procedure SCANNER($(A \rightarrow \alpha \bullet B\ \beta,\ [i,j])$)
  if B $\in$ PARTS-OF-SPEECH($word[j]$) **then**
    ENQUEUE($(B \rightarrow word[j] \bullet,\ [j, j+1])$, $chart[j+1]$)

- **برای همه غیرپایانه‌هایی مانند B که قبل از آن نقطه می‌آید:**
  - اگر آن غیرپایانه POS کلمه ورودی‌ای که در اندیس پایانی (j) باشد:
    - یک حالت جدید به چارت اضافه کنید که در آن نقطه به سمت راست قانون مرتبط منتقل شود.
    - اندیس نقطه را یک واحد اضافه کنید.

تجزیه Earley: گام‌های الگوریتم . . .

- کامل کننده (Completer) . . .

```
procedure COMPLETER((B → γ •, [j, k]))
    for each (A → α • B β, [i, j]) in chart[j] do
        ENQUEUE((A → α B • β, [i, k]), chart[k])
    end
```

- اگر نقطه عنصر پایانی یک قانون است که سمت چپ آن غیرپایانه B باشد (کامل شده):
  - قوانینی که نیازمند کامل شدن هستند و دارای عبارت غیرپایانه B هستند (نقطه قبل از B وجود دارد) و اندیس نقطه این قانون (j) و با اندیس عبارت یکی است:
    - قانون را به چارت اضافه کنید و نقطه را جابجا کنید (به بعد از B آورده شود)
    - اندیس نقطه با اندیش قانون کامل شده برابر قرار داده شود

تجزیه Earley

- الگوریتم

```
function EARLEY-PARSE(words, grammar) returns chart

  ADDTOCHART((γ → • S, [0,0]), chart[0])
  for i ← from 0 to LENGTH(words) do
    for each state in chart[i] do
      if INCOMPLETE?(state) and
              NEXT-CAT(state) is not a part of speech then
          PREDICTOR(state)
      elseif INCOMPLETE?(state) and
              NEXT-CAT(state) is a part of speech then
          SCANNER(state)
      else
          COMPLETER(state)
    end
  end
  return(chart)
```

```
procedure PREDICTOR((A → α • B β, [i, j]))
    for each (B → γ) in GRAMMAR-RULES-FOR(B, grammar) do
        ADDTOCHART((B → • γ, [j, j]), chart[j])
    end

procedure SCANNER((A → α • B β, [i, j]))
    if B ∈ PARTS-OF-SPEECH(word[j]) then
        ADDTOCHART((B → word[j] •, [j, j+1]), chart[j+1])

procedure COMPLETER((B → γ •, [j,k]))
    for each (A → α • B β, [i, j]) in chart[j] do
        ADDTOCHART((A → α B • β, [i,k]), chart[k])
    end

procedure ADDTOCHART(state, chart-entry)
    if state is not already in chart-entry then
        PUSH-ON-END(state, chart-entry)
    end
```

تجزیه Earley: مثال . . .

- ورودی
  - Mary feeds the otter
- مقداردهی اولیه
  - جدولی 5*5
  - **0** Mary **1** feeds **2** the **3** otter **4** eos **5**

تجزیه Earley: مثال . . .

0 Mary 1 feeds 2 the 3 otter 4 eos 5

Initialize

	0	1	2	3	4	5
0	X → • S eos					
1						
2						
3						
4						
5						

تجزیه Earley: مثال . . .

0 Mary 1 feeds 2 the 3 otter 4 eos 5

Predict

	0	1	2	3	4	5
0	X → • S eos **S → • NP VP** **NP → • N** **NP → • DET N** **N → • Mary** **N → • otter** **DET → • the**					
1						
2						
3						
4						
5						

procedure PREDICTOR($(A \rightarrow \alpha \bullet B\ \beta,\ [i,j])$)
  for each $(B \rightarrow \gamma)$ **in** GRAMMAR-RULES-FOR(B, *grammar*)
do
    ENQUEUE($(B \rightarrow \bullet\ \gamma,\ [j,j])$, *chart*[*j*])
  end

تجزیه Earley: مثال . . .

0 Mary 1 feeds 2 the 3 otter 4 eos 5

Scan

	0	1	2	3	4	5
0	X → • S eos S → • NP VP NP → • N NP → • DET N N → • Mary N → • otter DET → • the	**N → Mary •**				
1						
2						
3						
4						
5						

procedure SCANNER($(A \rightarrow \alpha \bullet B\ \beta,\ [i,j])$)
  if B $\in$ PARTS-OF-SPEECH($word[j]$) **then**
    ENQUEUE($(B \rightarrow word[j] \bullet,\ [j, j+1])$, $chart[j+1]$)

تجزیه Earley. مثال . . .

0 Mary 1 feeds 2 the 3 otter 4 eos 5

Complete

	0	1	2	3	4	5
0	X → • S eos S → • NP VP NP → • N NP → • DET N N → • Mary N → • otter DET → • the	N → Mary • **NP → N •** **S → NP • VP**				
1						
2						
3						
4						
5						

```
procedure COMPLETER((B → γ •, [j, k]))
    for each (A → α • B β, [i, j]) in chart[j] do
        ENQUEUE((A → α B • β, [i, k]), chart[k])
    end
```

تجزیه Earley. مثال . . .

0 Mary 1 feeds 2 the 3 otter 4 eos 5

Predict

	0	1	2	3	4	5
0	X → • S eos S → • NP VP NP → • N NP → • DET N N → • Mary N → • otter DET → • the	N → Mary • NP → N • S → NP • VP				
1		**VP → • V NP** **V → • feeds**				
2						
3						
4						
5						

تجزیه Earley. مثال . . .

0 Mary 1 feeds 2 the 3 otter 4 eos 5

Scan

	0	1	2	3	4	5
0	X → • S eos S → • NP VP NP → • N NP → • DET N N → • Mary N → • otter DET → • the	N → Mary • NP → N • S → NP • VP				
1		VP → • V NP V → • feeds	**V → feeds •**			
2						
3						
4						
5						

تجزیه Earley. مثال . . .

0 Mary 1 feeds 2 the 3 otter 4 eos 5

Complete

	0	1	2	3	4	5
0	X → • S eos S → • NP VP NP → • N NP → • DET N N → • Mary N → • otter DET → • the	N → Mary • NP → N • S → NP • VP				
1		VP → • V NP V → • feeds	V → feeds • **VP → V • NP**			
2						
3						
4						
5						

تجزیه Earley. مثال . . .

0 Mary 1 feeds 2 the 3 otter 4 eos 5

Predict

	0	1	2	3	4	5
0	X → • S eos S → • NP VP NP → • N NP → • DET N N → • Mary N → • otter DET → • the	N → Mary • NP → N • S → NP • VP				
1		VP → • V NP V → • feeds	V → feeds • VP → V • NP			
2			**NP → • N** **NP → • DET N** **N → • Mary** **N → • otter** **DET → • the**			
3						
4						
5						

تجزیه Earley. مثال . . .

0 Mary 1 feeds 2 the 3 otter 4 eos 5

Scan

	0	1	2	3	4	5
0	X → • S eos S → • NP VP NP → • N NP → • DET N N → • Mary N → • otter DET → • the	N → Mary • NP → N • S → NP • VP				
1		VP → • V NP V → • feeds	V → feeds • VP → V • NP			
2			NP → • N NP → • DET N N → • Mary N → • otter DET → • the	**DET → the •**		
3						
4						
5						

تجزیه Earley. مثال . . .

0 Mary 1 feeds 2 the 3 otter 4 eos 5

Complete

	0	1	2	3	4	5
0	X → • S eos S → • NP VP NP → • N NP → • DET N N → • Mary N → • otter DET → • the	N → Mary • NP → N • S → NP • VP				
1		VP → • V NP V → • feeds	V → feeds • VP → V • NP			
2			NP → • N NP → • DET N N → • Mary N → • otter DET → • the	DET → the • **NP → DET • N**		
3						
4						
5						

تجزیه Earley. مثال . . .

0 Mary 1 feeds 2 the 3 otter 4 eos 5

Predict

	0	1	2	3	4	5
0	X → • S eos S → • NP VP NP → • N NP → • DET N N → • Mary N → • otter DET → • the	N → Mary • NP → N • S → NP • VP				
1		VP → • V NP V → • feeds	V → feeds • VP → V • NP			
2			NP → • N NP → • DET N N → • Mary N → • otter DET → • the	DET → the • NP → DET • N		
3				**N → • Mary** **N → • otter**		
4						
5						

تجزیه Earley: مثال . . .

0 Mary 1 feeds 2 the 3 otter 4 eos 5

Scan

	0	1	2	3	4	5
0	X → • S eos S → • NP VP NP → • N NP → • DET N N → • Mary N → • otter DET → • the	N → Mary • NP → N • S → NP • VP				
1		VP → • V NP V → • feeds	V → feeds • VP → V • NP			
2			NP → • N NP → • DET N N → • Mary N → • otter DET → • the	DET → the • NP → DET • N		
3				N → • Mary N → • otter	**N → otter •**	
4						
5						

تجزیه Earley. مثال . . .

0 Mary 1 feeds 2 the 3 otter 4 eos 5

Complete

	0	1	2	3	4	5
0	X → • S eos S → • NP VP NP → • N NP → • DET N N → • Mary N → • otter DET → • the	N → Mary • NP → N • S → NP • VP				
1		VP → • V NP V → • feeds	V → feeds • VP → V • NP			
2			NP → • N NP → • DET N N → • Mary N → • otter DET → • the	DET → the • NP → DET • N	**NP → DET N •**	
3				N → • Mary N → • otter	N → otter •	
4						
5						

تجزیه Earley. مثال . . .

0 Mary 1 feeds 2 the 3 otter 4 eos 5

Complete

	0	1	2	3	4	5
0	X → • S eos S → • NP VP NP → • N NP → • DET N N → • Mary N → • otter DET → • the	N → Mary • NP → N • S → NP • VP				
1		VP → • V NP V → • feeds	V → feeds • VP → V • NP		**VP → V NP •**	
2			NP → • N NP → • DET N N → • Mary N → • otter DET → • the	DET → the • NP → DET • N	NP → DET N •	
3				N → • Mary N → • otter	N → otter •	
4						
5						

تجزیه Earley. مثال . . .

0 Mary 1 feeds 2 the 3 otter 4 eos 5

Complete

	0	1	2	3	4	5
0	X → • S eos S → • NP VP NP → • N NP → • DET N N → • Mary N → • otter DET → • the	N → Mary • NP → N • S → NP • VP			**S → NP VP •**	
1		VP → • V NP V → • feeds	V → feeds • VP → V • NP		VP → V NP •	
2			NP → • N NP → • DET N N → • Mary N → • otter DET → • the	DET → the • NP → DET • N	NP → DET N •	
3				N → • Mary N → • otter	N → otter •	
4						
5						

تجزیه Earley. مثال . . .

0 Mary 1 feeds 2 the 3 otter 4 eos 5

Complete

	0	1	2	3	4	5
0	X → • S eos S → • NP VP NP → • N NP → • DET N N → • Mary N → • otter DET → • the	N → Mary • NP → N • S → NP • VP			S → NP VP • **X → S • eos**	
1		VP → • V NP V → • feeds	V → feeds • VP → V • NP		VP → V NP •	
2			NP → • N NP → • DET N N → • Mary N → • otter DET → • the	DET → the • NP → DET • N	NP → DET N •	
3				N → • Mary N → • otter	N → otter •	
4						
5						

تجزیه Earley: مثال . . .

0 Mary 1 feeds 2 the 3 otter 4 eos 5

Predict

	0	1	2	3	4	5
0	X → • S eos S → • NP VP NP → • N NP → • DET N N → • Mary N → • otter DET → • the	N → Mary • NP → N • S → NP • VP			S → NP VP • X → S • eos	
1		VP → • V NP V → • feeds	V → feeds • VP → V • NP		VP → V NP •	
2			NP → • N NP → • DET N N → • Mary N → • otter DET → • the	DET → the • NP → DET • N	NP → DET N •	
3				N → • Mary N → • otter	N → otter •	
4					eos → • eos	
5						

تجزیه Earley. مثال . . .

0 Mary 1 feeds 2 the 3 otter 4 eos 5

Scan

	0	1	2	3	4	5
0	X → • S eos S → • NP VP NP → • N NP → • DET N N → • Mary N → • otter DET → • the	N → Mary • NP → N • S → NP • VP			S → NP VP • X → S • eos	
1		VP → • V NP V → • feeds	V → feeds • VP → V • NP		VP → V NP •	
2			NP → • N NP → • DET N N → • Mary N → • otter DET → • the	DET → the • NP → DET • N	NP → DET N •	
3				N → • Mary N → • otter	N → otter •	
4					eos → • eos	**eos → eos •**
5						

تجزیه Earley. مثال

0 Mary 1 feeds 2 the 3 otter 4 eos 5

Complete

	0	1	2	3	4	5
0	X → • S eos S → • NP VP NP → • N NP → • DET N N → • Mary N → • otter DET → • the	N → Mary • NP → N • S → NP • VP			S → NP VP • X → S • eos	**X → S eos •**
1		VP → • V NP V → • feeds	V → feeds • VP → V • NP		VP → V NP •	
2			NP → • N NP → • DET N N → • Mary N → • otter DET → • the	DET → the • NP → DET • N	NP → DET N •	
3				N → • Mary N → • otter	N → otter •	
4					eos → • eos	eos → eos •
5						

تجزیه Earley: مثال ۲ . . .

- جمله
  - Book That Flight

تجزیه Earley: مثال ۲ . . .

- Chart[0]

S0	$\gamma \rightarrow \bullet S$	[0,0]	Dummy start state
S1	$S \rightarrow \bullet NP\ VP$	[0,0]	Predictor
S2	$S \rightarrow \bullet Aux\ NP\ VP$	[0,0]	Predictor
S3	$S \rightarrow \bullet VP$	[0,0]	Predictor
S4	$NP \rightarrow \bullet Pronoun$	[0,0]	Predictor
S5	$NP \rightarrow \bullet Proper\text{-}Noun$	[0,0]	Predictor
S6	$NP \rightarrow \bullet Det\ Nominal$	[0,0]	Predictor
S7	$VP \rightarrow \bullet Verb$	[0,0]	Predictor
S8	$VP \rightarrow \bullet Verb\ NP$	[0,0]	Predictor
S9	$VP \rightarrow \bullet Verb\ NP\ PP$	[0,0]	Predictor
S10	$VP \rightarrow \bullet Verb\ PP$	[0,0]	Predictor
S11	$VP \rightarrow \bullet VP\ PP$	[0,0]	Predictor

برای یک گرامر، این مرحله به ازای همه ورودی ها یکسان است و می تواند یک بار بارگذاری شود

تجزیه Earley. مثال ۲ . . .

- Chart[1]

S12	*Verb* → *book* •	[0,1]	Scanner
S13	*VP* → *Verb* •	[0,1]	Completer
S14	*VP* → *Verb* • *NP*	[0,1]	Completer
S15	*VP* → *Verb* • *NP PP*	[0,1]	Completer
S16	*VP* → *Verb* • *PP*	[0,1]	Completer
S17	*S* → *VP* •	[0,1]	Completer
S18	*VP* → *VP* • *PP*	[0,1]	Completer
S19	*NP* → • *Pronoun*	[1,1]	Predictor
S20	*NP* → • *Proper-Noun*	[1,1]	Predictor
S21	*NP* → • *Det Nominal*	[1,1]	Predictor
S22	*PP* → • *Prep NP*	[1,1]	Predictor

تجزیه Earley. مثال ۲

- Chart[2] و Chart[3]

S23	*Det* → *that* •	[1,2]	Scanner
S24	*NP* → *Det* • *Nominal*	[1,2]	Completer
S25	*Nominal* → • *Noun*	[2,2]	Predictor
S26	*Nominal* → • *Nominal Noun*	[2,2]	Predictor
S27	*Nominal* → • *Nominal PP*	[2,2]	Predictor
S28	*Noun* → *flight* •	[2,3]	Scanner
S29	*Nominal* → *Noun* •	[2,3]	Completer
S30	*NP* → *Det Nominal* •	[1,3]	Completer
S31	*Nominal* → *Nominal* • *Noun*	[2,3]	Completer
S32	*Nominal* → *Nominal* • *PP*	[2,3]	Completer
S33	*VP* → *Verb NP* •	[0,3]	Completer
S34	*VP* → *Verb NP* • *PP*	[0,3]	Completer
S35	*PP* → • *Prep NP*	[3,3]	Predictor
S36	*S* → *VP* •	[0,3]	Completer
S37	*VP* → *VP* • *PP*	[0,3]	Completer