**دانشکده‌ی علوم ریاضی**

**نظریه‌ی زبان‌ها و اتوماتا** ۲۱ آبان ۱۳۹۱

# جلسه‌ی ۱۷: بررسی گرامرهای مبهم، فرم نرمال چامسکی

*مدرّس: دکتر شهرام خزائی* *نگارندگان: حسین حاجی سلطانی، مهتاب کریمی و سینا مسندی*

## ۱ گرامرهای مبهم

**تعریف ۱** *زبان مستقل از متن $L$ را ذاتاً مبهم می‌گوییم اگر هر گرامر آن مبهم باشد.*

**قضیه ۱** *برای هر گرامر مستقل از متن $G = (V, T, P, S)$ رشته $w$ از پایانه‌ها دارای دو درخت تجزیه متمایز است، اگر و فقط اگر $w$ دارای دو اشتقاق چپ‌ترین از متغیر شروع باشد .*

**مثال ۱** *گرامر زیر را در نظر بگیرید:*

$$
\begin{aligned}
&E \to I|E+E|E*E|(E) \\
&I \to a|b|Ia|Ib|I1|I0
\end{aligned}
$$

*نشان دهید این گرامر مبهم است.*

*$w = a + a * a$ را در نظر می‌گیریم، $w$ در این زبان پذیرفته است.*
*زبان این گرامر مبهم است، زیرا $w$ دارای دو اشتقاق چپ‌ترین از متغیر شروع می‌باشد:*

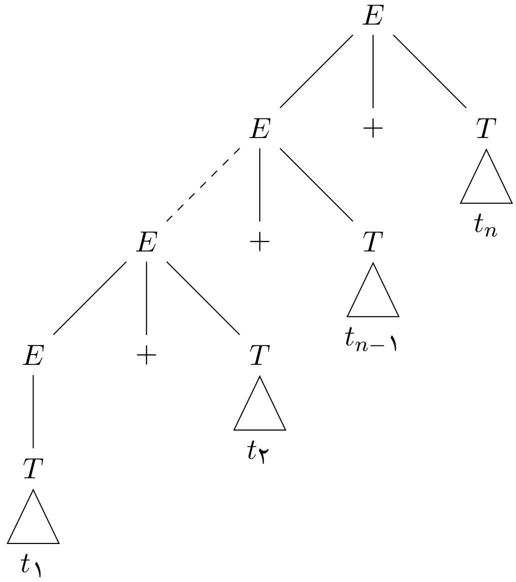
$$
\begin{aligned}
E &\underset{lm}{\Rightarrow} E*E \underset{lm}{\Rightarrow} E+E*E \underset{lm}{\Rightarrow} I+E*E \underset{lm}{\Rightarrow} a+E*E \\
&\underset{lm}{\Rightarrow} a+I*E \underset{lm}{\Rightarrow} a+a*E \underset{lm}{\Rightarrow} a+a*I \underset{lm}{\Rightarrow} a+a*a \\
E &\underset{lm}{\Rightarrow} E+E \underset{lm}{\Rightarrow} I+E \underset{lm}{\Rightarrow} a+E \underset{lm}{\Rightarrow} a+E*E \\
&\underset{lm}{\Rightarrow} a+I*E \underset{lm}{\Rightarrow} a+a*E \underset{lm}{\Rightarrow} a+a*I \underset{lm}{\Rightarrow} a+a*a
\end{aligned}
$$

**مثال ۲** *زبان گرامر مثال قبل را در نظر بگیرید، نشان دهید این زبان ذاتاً مبهم نیست.*
*یک گرامر نامبهم برای این زبان می‌سازیم:*

**گزاره ۱** *یک عبارت حاصل جمع تعدادی جمع وند[1] است.*

$$E = t_1 + t_2 + \cdots + t$$

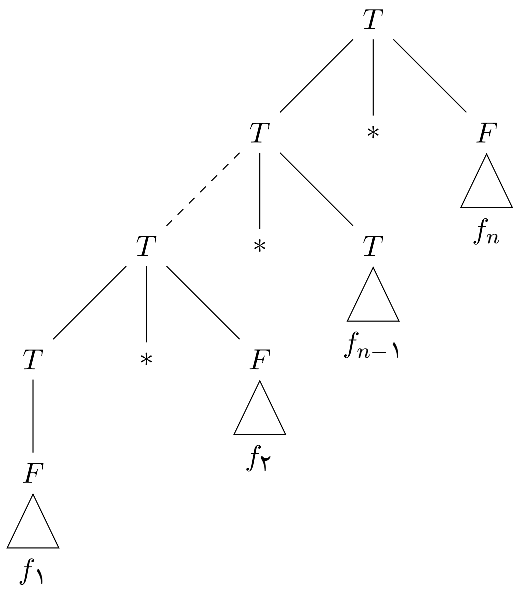
[1] term




مثلاً عبارت $\underline{a}+\underline{a*b}+\underline{c*(a*b)}$ دارای سه جمع وند مشخص شده است.

**گزاره ۲** یک جمع وند حاصل ضرب تعدادی عامل[2] است.

$$t = f_1 * f_2 * \ldots * f_n$$




[2]factor

*گرامر جدید را به این صورت تعریف می‌کنیم:*

$$\begin{aligned}
&E \to E+T|T\\
&T \to F|T*F\\
&F \to I|(E)\\
&I \to a|b|Ia|Ib|I1|I0
\end{aligned}$$

**مثال ۳** *زبان مستقل از متن زیر را در نظر بگیرید:*

$$L=\{0^i1^j2^k|i,j,k\geq 0 \wedge (i=j \vee j=k)\}$$

*نشان دهید این زبان ذاتاً مبهم نیست.*

*گرامر زیر را برای آن به گونه ای در نظر می‌گیریم که مبهم نباشد:*

$$\begin{aligned}
&L=\{0^n1^n2^m|n,m\geq 0\} \vee \{0^m1^n2^n|n,m\geq 0\}\\
&A:\{0^n1^n\} \qquad B:\{2^m\} \qquad C:\{0^m\} \qquad D:\{1^n2^n\}\\
&S \to AB|CD\\
&A \to 0A1|\epsilon\\
&D \to 1D2|\epsilon\\
&B \to 2B|\epsilon\\
&C \to 0C|\epsilon
\end{aligned}$$

*گرامر فوق بدون ابهام است، پس این زبان ذاتاً مبهم نیست.*

## ۲ فرم نرمال چامسکی

**تعریف ۲** *(فرم نرمال چامسکی[۳]) گرامری که هر قانون تولید آن به یکی از دو صورت زیر باشد،یک گرامر به فرم نرمال چامسکی نامیده می‌شود:*

$$\begin{aligned}
A &\to a\\
A &\to BC
\end{aligned}$$

در این جلسه نشان می‌دهیم که برای هر زبان مستقل از متن $L$، یک گرامر به فرم نرمال چامسکی برای $L-\{\epsilon\}$ وجود دارد. برای رسیدن به فرم نرمال چامسکی به یک روند تمیزکاری برای گرامر نیاز داریم که به ترتیب زیر اعمال گفته شده را انجام دهد:

۱. قوانین تولید $\epsilon$ را حذف کند.

۲. قوانین تولید یکه را حذف کند.

۳. متغیرهای بی‌استفاده را حذف کند.

برای این کار ابتدا به ارائه‌ی تعاریف و الگوریتم‌هایی که در ادامه معرفی می‌شوند نیازمندیم.

[۳] Chomsky Normal Form

## ۳ حذف قوانین تولید $\epsilon$

هر قانون به صورت $A \to \epsilon$ یک قانون $\epsilon$[۴] نامیده می‌شود. برای حذف قوانین تولید $\epsilon$ از یک گرامر بطوری که تنها رشته تهی از زبان آن حذف شود نیاز به تعریف زیر داریم. در جلسه بعد درباره حذف قوانین $\epsilon$ صحبت خواهیم کرد.

### ۱.۳ متغیر تهی‌ساز و الگوریتم کشف آنها

**تعریف ۳** (*متغیر تهی‌ساز*[۵]) *متغیر $A$ را تهی‌ساز گوییم اگر $A \overset{*}{\Rightarrow} \epsilon$.*

الگوریتم بازگشتی زیر مجموعه متغیرهای تهی‌ساز را پیدا می‌کند:

پایه: اگر $A \to \epsilon$ یک قانون تولید باشد، $A$ یک متغیر تهی‌ساز است.

استقرا: اگر $A \to \alpha$ یک قانون تولید باشد که $\alpha \in V^*$ و همه‌ی متغیرهای $\alpha$ تهی‌ساز باشند، $A$ نیز تهی‌ساز است.

**مثال ۴** *گرامر $G$ را به صورت زیر در نظر بگیرید:*

$$\begin{cases} S \to ABC \\ A \to aA|\epsilon \\ B \to bB|\epsilon \\ C \to \epsilon \end{cases}$$

*مجموعه متغیرهای تهی‌ساز گرامر $G$ را با $n(G)$ نشان می‌دهیم. داریم:*

*پایه:* $n(G) = \{A, B, C\}$

*استقرا:* $n(G) = \{S, A, B, C\}$

*بنابراین همه متغیرها تهی‌ساز هستند. در جلسه بعد درباره حذف قوانین $\epsilon$ صحبت خواهیم کرد که گرامر را به صورت زیر در می‌آورد:*

$$\begin{cases} S \to ABC \\ A \to aA|\epsilon \\ B \to bB|\epsilon \\ C \to \epsilon \end{cases} \to \begin{cases} S \to ABC|BC|AC|AB|A|B|C|\epsilon \\ A \to aA|a|\epsilon \\ B \to bB|b|\epsilon \\ C \to \epsilon \end{cases} \to \begin{cases} S \to ABC|BC|AC|AB|A|B|C \\ A \to aA|a \\ B \to bB|b \end{cases}$$

## ۴ حذف قوانین تولید یکه

هر قانون تولید به صورت $A \to B$ یک قانون تولید یکه[۶] نامیده می‌شود. روش حذف قوانین تولید یکه در جلسه بعد بررسی می‌شود.

---

[۴] $\epsilon$-producion

[۵] nullable variable

[۶] unit production

## ۵ متغیرهای بی‌استفاده و الگوریتم کشف آنها

**تعریف ۴** *(متغیر مفید[۷] و متغیر بی‌استفاده[۸]) متغیر $A$ برای گرامر $G = (V, T, P, S)$ مفید نامیده می‌شود اگر یک اشتقاق به صورت زیر وجود داشته باشد:*

$$S \overset{*}{\Rightarrow} \alpha A \beta \overset{*}{\Rightarrow} w$$

*که در آن*

$$w \in T^* \text{ و } \alpha, \beta \in (V \cup T)^*.$$

*در غیر اینصورت متغیر بی‌استفاده نامیده می‌شود.*

**قضیه ۲** *فرض کنید $G$ یک گرامر مستقل از متن باشد، گرامری که پس از انجام دو مرحله‌ی زیر به دست می‌آید، همان زبان را تعریف می‌کند و دارای متغیر بی‌استفاده نیست.*

۱- *حذف همه‌ی متغیرهای غیرمولد و تمام قوانینی که در آنها ظاهر می‌شوند.*

۲- *حذف همه‌ی متغیرهای غیرقابل دسترس و تمام قوانینی که در آن‌ها ظاهر می‌شوند.*

در ادامه به معرفی متغیر مولد و متغیر قابل دسترس و الگوریتم‌هایی برای پیدا کردن آنها می‌پردازیم.

### ۱.۵ متغیر مولد و الگوریتم کشف آنها

**تعریف ۵** *(متغیر مولد[۹]) متغیر $A$ مولد نامیده می‌شود اگر برای حداقل یک رشته $w$ داشته باشیم $A \overset{*}{\Rightarrow} w$.*

الگوریتم بازگشتی زیر مجموعه متغیرهای مولد را پیدا می‌کند:

پایه: اگر $A \to w$ یک قانون تولید باشد، آنگاه $A$ مولد است.

استقرا: اگر $A \to \alpha$ یکی از قوانین تولید باشد و همه‌ی متغیرهای رشته‌ی $\alpha$ مولد باشند، $A$ نیز مولد است.

**مثال ۵** *برای مثال در ادامه روند تبدیل گرامر زیر را به فرم نرمال چامسکی می‌بینیم:*

$$\begin{cases} S \to AB|C \\ A \to aA|a \\ B \to bB \\ C \to c \end{cases}$$

[۷] useful variable

[۸] useless variable

[۹] generating variable

*مجموعه متغیرهای مولد گرامر $G$ را با $g(G)$ نشان می‌دهیم. داریم:*

*پایه:* $g(G) = \{A, C\}$

*استقرا:* $g(G) = \{A, C, S\}$

*و لذا تنها متغیر غیر مولد $B$ است که پس از حذف آن، گرامر به صورت زیر در می‌آید:*

$$\begin{cases} S \to AB|C \\ A \to aA|a \\ B \to bB \\ C \to c \end{cases} \to \begin{cases} S \to AB|C \\ A \to aA|a \\ C \to c \end{cases} \to \begin{cases} S \to C \\ A \to aA|a \\ C \to c \end{cases}$$

## ۲.۵ متغیر قابل دسترس و الگوریتم کشف آنها

**تعریف ۶** *(متغیر قابل دسترس[۱۰]) متغیر $A$ قابل دسترس نامیده می‌شود اگر رشته‌های $\alpha, \beta \in (V \cup T)^*$ وجود داشته باشد که $S \overset{*}{\Rightarrow} \alpha A\beta$.*

الگوریتم بازگشتی زیر مجموعه متغیرهای قابل دسترس را پیدا می‌کند:

پایه: $S$ قابل دسترس است.

استقرا: اگر $A$ قابل دسترس باشد و $A \to \alpha$ یک قانون تولید باشد، تمام متغیرهای $\alpha$ نیز قابل دسترس هستند.

**مثال ۶** *در ادامه روند تبدیل گرامر مثال قبل به فرم نرمال چامسکی که پس از حذف متغیرهای غیر مولد بصورت زیر در آمد،*

$$\begin{cases} S \to AB|C \\ A \to aA|a \\ C \to c \end{cases}$$

*مجموعه متغیرهای مولد گرامر $G$ را با $r(G)$ نشان می‌دهیم. داریم:*

*پایه:* $r(G) = \{S\}$

*استقرا:* $r(G) = \{C, S\}$

*لذا تنها متغیر غیر مولد $A$ است که پس از حذف آن، گرامر به صورت زیر در می‌آید:*

$$\begin{cases} S \to C \\ A \to aA|a \\ C \to c \end{cases} \to \begin{cases} S \to C \\ C \to c \end{cases}$$

*گرامر حاصل فقط دارای متغیرهای مفید است.*

---
[۱۰] reachable variable