

اصول طراحی کامپایلر

گروه کامپیوتر و دانشگاه آزاد اسلامی واحد زنجان

مدرس: مهدی بازرگانی
آخرین بروز رسانی: پاییز ۱۳۹۲

Zau-93-94-2

فصل دوم

دسته بندی گرامرها (یادآوری)

Zau-93-94-2

مثالی از یک گرامر زبان طبیعی

- ❖ گرامر:
بایدها و نبایدهای یک زبان، شکل صحیح نوشتار یک جمله را به ما بیان می کند.
- ❖ به بیان دیگر گرامر تعریف یک زبان را شامل می شود.
- ❖ (۱) جمله ← جمله خبری | جمله پرسشی | جمله امری | جمله تعجبی
- ❖ (۲) جمله خبری ← فاعل + مفعول + "را" + فعل
- ❖ (۳) فاعل ← "علی" | "حسن" | اسم
- ❖ (۴) مفعول ← "کتاب" | "مدرسه" | "غذا" | اسم
- ❖ (۵) فعل ← "می رود" | "خورد" | "می خورد"
- ❖ (۶) علی کتاب را می خواند. لا بر طبق گرامر است
- ❖ علی کتاب را خورد.
- ❖ از نظر syntax (نحو) درست از نظر semantic (معنا) نادرست میباشد

Zau-93-94-2

وظیفه گرامر

❖ وظیفه یک گرامر تولید جمله می باشد. شامل ۴ تایی مرتب است.

$G = (N, T, S, P)$

مجموعه قواعد تولید (Production Rules)

سمبل آغاز (start symbol)

مجموعه پایانی ها (Terminal set)

مجموعه غیر پایانی ها (Non-terminal set)

Zau-93-94-2

مثالی از یک گرامر زبان صوری

❖ وظیفه یک گرامر تولید جمله می باشد.

$$N = \{ E, F \}$$

$$T = \{ +, *, /, id \}$$

$$S = E$$

$$P = \{ E \rightarrow F * id, F \rightarrow F / E, F \rightarrow F + F | id \}$$

رشته تولیدی نمونه $\leftarrow$ $id + id * id$

Zau-93-94-2

اشتقاق (Derivation)

❖ فرآیند تولید رشته از گرامر با شروع از عنصر ابتدای گرامر و استفاده از قوانین.

از چپ: در هر قدم انجام جایگزینی روی سمت چپ ترین غیر پایانی

از راست: در هر قدم انجام جایگزینی روی سمت راست ترین غیر پایانی

انواع اشتقاق

Zau-93-94-2

مثال از اشتقاق

تولید رشته $id + id * id$

اشتقاق چپ

$E \rightarrow E + E$
 $E \rightarrow id + E$
 $E \rightarrow id + E * E$
 $E \rightarrow id + id * E$
 $E \rightarrow id + id * id$

اشتقاق راست

$E \rightarrow E + E$
 $E \rightarrow E + E * E$
 $E \rightarrow E + E * id$
 $E \rightarrow E + id * id$
 $E \rightarrow id + id * id$

Zau-93-94-2

درخت تجزیه

❖ درخت تجزیه نشان دهنده چگونگی اشتقاق رشته ای از زبان

از نماد شروع گرامر



❖ ساخت درخت تجزیه:

ریشه درخت S -

قانون $A = A \rightarrow XYZ$ اگره ای در درخت و XYZ فرزندان آن

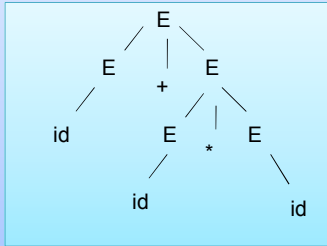
قانون $A = A \rightarrow Xa$ اگره ای در درخت و X ,a فرزندان آن

پایانه ها (حروف کوچک) تنها در برگها دیده می شوند.

Zau-93-94-2

درخت اشتقاق

❖ درخت تجزیه ای نشان دهنده مراحل اشتقاق بکار رفته است.
(اشتقاق راست یا اشتقاق چپ)



مثال

$id + id * id$

Zau-93-94-2

گرامر مبهم



❖ وجود دو اشتقاق راست یا
❖ دو اشتقاق چپ برای یک رشته در گرامر
❖ مثال $id + id * id$
اشتقاق چپ اول

اشتقاق چپ دوم

$E \Rightarrow E+E$
 $E \Rightarrow id+E$
 $E \Rightarrow id+E * E$
 $E \Rightarrow id+id * E$
 $E \Rightarrow id+id * id$

$E \Rightarrow E * E$
 $E \Rightarrow E + E * E$
 $E \Rightarrow id + E * E$
 $E \Rightarrow id + id * E$
 $E \Rightarrow id + id * id$

Zau-93-94-2

دسته بندی چامسکی



نکته ۱: تفاوت گرامر ها در قوانین تولید (P) آنهاست.

نکته ۲: محدودیت گرامر ها از نوع منظم به بعد کمتر می شود.

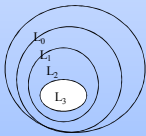
نکته ۳: نامقید ترین نوع گرامر نوع صفر است که هیچ شرطی بر قوانین تولید آن اعمال نمی شود به بیان دیگر در این گرامر هر ترکیبی از پایانی ها و غیر پایانی ها می توانند در سمت چپ و راست خود داشته باشند. تنها محدودیت این نوع گرامر این است که

❖ نمی تواند در سمت چپ یک قانون باشد.

Zau-93-94-2

دسته بندی چامسکی

نکته ۵: گرامر های نامقید قوی تر از گرامر های منظم و مستقل از متن هستند در واقع گرامر های نامقید بزرگترین خانواده زبانهایی را ایجاد می کنند که می خواهیم به واسطه مکانیکی آنها را تشخیص دهیم.



Zau-93-94-2

دسته بندی چامسکی گرامرهای صوری

$A \rightarrow aB|a$ خطی از راست

۱- گرامر نوع سوم یا منظم یا خطی: Regular Grammar

$A \rightarrow Ba|a$ خطی از چپ

۲- گرامر نوع دوم یا مستقل از متن context free Grammar

$A \rightarrow \alpha; \alpha \in (V \cup T)^*$

۳- گرامر نوع اول یا حساس به متن context sensitive Grammar

$\alpha \rightarrow \beta; |\alpha| \leq |\beta|; \beta \in (V \cup T)^*; \alpha \in (V \cup T)^+$

۴- گرامر نوع صفرم یا نامنظم (نامقید) Irregular Grammar

$\alpha \rightarrow \beta; \beta \in (V \cup T)^*; \alpha \in (V \cup T)^+$

Zau-93-94-2

۱۳

ساده سازی گرامرهای نوع دوم

۱- غیرپایانی غیر قابل دسترس (بلا استفاده) و قواعدی که این غیرپایانی ها در آن استفاده شده است را حذف می کنیم.

۲- قواعد واحد را حذف می نماییم. اثر قواعد واحد در دل قواعد دیگر باید انجام گیرد.

۳- غیرپایانی های پوچ را حذف می کنیم.

Zau-93-94-2

نرماسازی گرامرهای مستقل از متن

۱- حذف قاعده بازگشتی از چپ:

a. قوانین را به دودسته بازگشتی از چپ وبدون بازگشتی از چپ تقسیم میکنیم.

b. متغیر جدید تعریف میکنیم وباستفاده از قوانین فاقد بازگشتی از چپ قوانین دارای بازگشتی از چپ را حذف میکنیم.

Zau-93-94-2

نرماسازی گرامرهای مستقل از متن (مثال...)

$A \rightarrow Aa|Ab|Ac|X|Y$

$A \rightarrow Aa|Ab|Ac$

$A \rightarrow X|Y$

$A \rightarrow XZ|YZ$

$Z \rightarrow aZ|bZ|cZ$

Zau-93-94-2

نرماسازی گرامرهای مستقل از متن

۲- فاکتورگیری از چپ:

نوعی تبدیل گرامر است که باعث این میشود به عنوان یک گرامر مناسب در پاسرهای پیشگوپردازش گردد.
حذف قواعد بلا استفاده، حذف قواعد واحد و حذف قواعد پوچ از کتاب نظریه زبانها و ماشینها فصل ششم مطالعه گردد.

Zau-93-94-2

گرامرهای مستقل از متن

آیروشی وجود دارد که از یک گرامر مستقل از متن رفع ابهام کرد؟

-گرامرهایی داریم که ذاتا مبهم هستند این گرامرها تحت هیچ شرایطی رفع ابهام نخواهند شد.

-اما گرامرهایی که ذاتا مبهم نیستند با تغییر و تعریف متغیرهای جدید میتوان ابهامشان را رفع کرد.

Zau-93-94-2

عملیات زبانهای رسمی

OPERATION	DEFINITION
<i>union of L and M written $L \cup M$</i>	$L \cup M = \{s \mid s \text{ is in } L \text{ or } s \text{ is in } M\}$
<i>concatenation of L and M written LM</i>	$LM = \{st \mid s \text{ is in } L \text{ and } t \text{ is in } M\}$
<i>Kleene closure of L written L^*</i>	$L^* = \bigcup_{i=0}^{\infty} L^i$ L^* denotes zero or more concatenations of L
<i>positive closure of L written L^+</i>	$L^+ = \bigcup_{i=1}^{\infty} L^i$ L^+ denotes one or more concatenations of L

Zau-93-94-2

مثال

$$L = \{A, B, C, D\} \quad D = \{1, 2, 3\}$$

$$L \cup D = \{A, B, C, D, 1, 2, 3\}$$

$$LD = \{A1, A2, A3, B1, B2, B3, C1, C2, C3, D1, D2, D3\}$$

$$L^2 = \{AA, AB, AC, AD, BA, BB, BC, BD, CA, \epsilon, DD\}$$

$$L^4 = L^2 L^2 = ??$$

$$L^* = \{ \text{All possible strings of } L \text{ plus } \epsilon \}$$

$$L^+ = L^* - \epsilon$$

$$L(L \cup D) = ??$$

$$L(L \cup D)^* = ??$$

Zau-93-94-2

زبان های با قاعده (عبارات منظم RE)

دلالیت بر زبانهای نوع سوم دارند.
نماینده زبانهای منظم می باشند.
ارزش آنها معادل یک زبان منظم است.
ترکیبی از الفبای زبان Σ , +, *, (,) می باشند.

Zau-93-94-2

Rules

- الفبای ثابت Σ
- ϵ عبارت معین مبتنی بر $\{ \epsilon \}$ است
- اگر a در سیگما باشد عبارت معین مبتنی بر a به صورت زیر خواهد بود
- If a is in Σ , a is a regular expression that denotes $\{a\}$ if w, v, j, c, d, v
- Let r and s be R.E. for $L(r)$ and $L(s)$. Then
- (a) $(r) \mid (s)$ is a regular expression $L(r) \cup L(s)$
- (b) $(r)(s)$ is a regular expression $L(r) L(s)$
- (c) $(r)^*$ is a regular expression $(L(r))^*$
- (d) (r) is a regular expression $L(r)$
- All are Left-Associative.
- Parentheses are dropped as allowed by precedence.



Zau-93-94-2

مرور مجدد مثال

$L = \{A, B, C, D\}$ $D = \{1, 2, 3\}$

$A \mid B \mid C \mid D = L$
 $(A \mid B \mid C \mid D) (A \mid B \mid C \mid D) = L$
 $(A \mid B \mid C \mid D)^* = L^*$
 $(A \mid B \mid C \mid D) ((A \mid B \mid C \mid D) \mid (1 \mid 2 \mid 3)) = L(1D)$

Zau-93-94-2

خواص جبری

AXIOM	DESCRIPTION
$r \mid s = s \mid r$	$\mid$ is commutative
$r \mid (s \mid t) = (r \mid s) \mid t$	$\mid$ is associative
$(r \mid s) t = r (s \mid t)$	concatenation is associative
$r (s \mid t) = r s \mid r t$ $(s \mid t) r = s r \mid t r$	concatenation distributes over $\mid$
$\epsilon r = r$ $r \epsilon = r$	ϵ Is the identity element for concatenation
$r^* = (r \epsilon)^*$	relation between ϵ and $*$
$r^{**} = r^*$	$*$ is idempotent

Zau-93-94-2

مثال های بیشتر

- ❖ تمام رشته هایی که با "bat" شروع می شود یا با "tab" پایان می پذیرد
- ❖ $\text{bat}\{A,\dots,Z,a,\dots,z\}^* \mid \{A,\dots,Z,a,\dots,z\}^*\text{tab}$
- ❖ تمام رشته هایی از $\{1,2,3\}$ که به ترتیب صعودی مرتب شده اند
- ❖ $\{A,\dots,Z\}^*1\{A,\dots,Z\}^*2\{A,\dots,Z\}^*3\{A,\dots,Z\}^*$

Zau-93-94-2

مثال های بیشتر

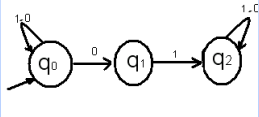
تعریف عبارات منظم مفهومی : def(semantic)

- 1. $\emptyset$ Represent $\{\}$
 - 2. λ Represent $\{\lambda\}$
 - 3. $a \in \Sigma$ Represent $\{a\}$
 - 4. if $R1, R2$ are R.E then
 $L(R1+R2)=L(R1) \cup L(R2)$
 $L(R1.R2)=L(R1) \cdot L(R2)$
 $L(R1^*)= (L(R1))^*$
 $(L(R1))= L(R1)$
- مثال:
- $(0+1)^*011^* \quad \Sigma=\{0,1\}$
 $0 \in \Sigma: \{0\} \quad 0.1:\{0\}.\{1\}=\{0.1\}$
 $1 \in \Sigma:\{1\} \quad 0+1:\{0\} \cup \{1\} \rightarrow (0+1):\{0,1\}$
 $(0+1)^*:\{0,1\}^*$
 $1^*:\{1\}^*$

Zau-93-94-2

مثال های بیشتر

- مثال:
- $(0+1)^*011^* \quad \Sigma=\{0,1\}$
- $0 \in \Sigma: \text{R.E} \rightarrow 0+1: \text{R.E} \rightarrow (0+1): \text{R.E}$
- $1 \in \Sigma: \text{R.E}$
- 1: $(0+1)^*: \text{R.E}$
2: $0.1: \text{R.E} \quad 1,2,3 \rightarrow (0+1)^*011^*: \text{R.E}$
3: $1^*: \text{R.E}$



Zau-93-94-2