اصول طراحی کامپایلر

درس ۱۴

تحلیل نحوی (۹)

برخورد با خطاهای نحوی در تجزیه‌گر

Syntax Analysis (9)

Error-Handling in Parser

کاظم فولادی قلعه

دانشکده مهندسی، پردیس فارابی

دانشگاه تهران

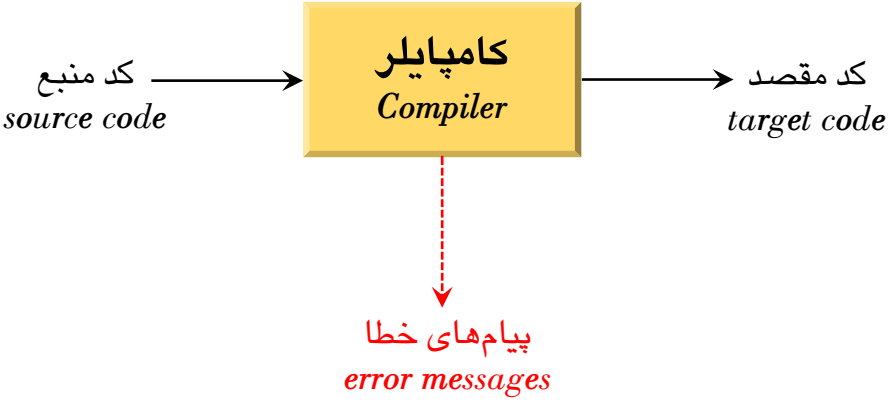
http://courses.fouladi.ir/compiler

تحلیل نحوی
برخورد با خطاهای نحوی در تجزیه‌گر

۱

مقدمه

اداره کردن خطاها توسط کامپایلر



انواع خطا در فرآیند کامپایل

محل رخداد خطا	نام خطا	مثال
تحلیل لغوی	**خطای لغوی (lexical error)**	شروع کردن یک شناسه با عدد
تحلیل نحوی	**خطای نحوی (syntax error)**	عدم تطابق پرانترها در یک عبارت ریاضی
تحلیل معنایی	**خطای معنایی (semantic error)**	انتساب یک مقدار اعشاری به متغیر صحیح
تولید کد میانی		کمبود حافظه بر روی دیسک
بهینه‌سازی کد میانی		کمبود حافظه بر روی دیسک
تولید کد نهایی		کمبود حافظه بر روی دیسک

خطای نحوی: خطایی که در سطح تحلیل نحوی شناسایی می‌شود.

* **اغلب، کشف خطا و ترمیم آن در مرحله‌ی تحلیل نحوی انجام می‌شود، زیرا:**
- **(ویژگی طبیعی خطا)** طبیعت بسیاری از خطاها نحوی است؛
- **(توانایی ویژه‌ی تجزیه‌گرها)** ویژگی‌های روش‌های تجزیه، می‌تواند خطاهای نحوی را با دقت بالایی کشف کند.

روی‌کردهای برخورد با خطا

برخورد با خطا

Error Handling

توقف با رسیدن به اولین خطا

Stopping

۱

ترمیم و گذر از خطا

Error Recovery

۲

برای کشف خطاهای بیشتر

در هر دو صورت، کد منبع دارای خطاست و قابل ترجمه به کد مقصد نیست!

گزارش خطا

گزارش خطا

Error Reporting

گزارش محل خطا در کد منبع

Error Location

۱

گزارش نوع خطا

Error Type

۲

انتظارات از ماژول برخورد با خطا در تجزیه‌گر

۱. گزارش دقیق و صریح وجود خطا
۲. گذر سریع از روی خطا به منظور کشف خطاهای بعدی
۳. لطمه نزدن به سرعت پردازش برنامه‌های صحیح
۴. * حدس زدن منظور برنامه‌نویس در محل خطا

تحلیل نحوی
برخورد با خطاهای نحوی در تجزیه‌گر

۲

استراتژی‌های گذر از خطا

استراتژی‌های گذر از خطا

۱. **استراتژی حالت وحشت**
Panic Mode Strategy

۲. **استراتژی سطح عبارت**
Phrase Level Strategy

۳. **استراتژی قواعد خطا**
Error Production Strategy

۴. **استراتژی تصحیح سراسری**
Global Correction Strategy

معمولاً ترکیبی از این استراتژی‌ها استفاده می‌شود!

استراتژی‌های گذر از خطا

استراتژی حالت وحشت

تجزیه‌گر در هنگام کشف خطا،
نمادهای ورودی را تا رسیدن به **یک توکن مناسب** دور می‌ریزد.

استراتژی حالت وحشت
Panic Mode Strategy

مثلاً:
- توکن پایان‌دهنده‌ی یک ساختار (**end** یا **;**)
- توکن آغازکننده‌ی یک ساختار (مثل کلمات کلیدی، **if**)

مزایا	معایب
ساده‌ترین استراتژی برای پیاده‌سازی	امکان حذف حجم زیادی از ورودی
قابل استفاده با بیشتر روش‌های تجزیه	
تضمین عدم گیرافتادن در حلقه‌ی بی‌پایان	
مناسب برای موارد وقوع چندین خطا در یک دستور به‌ندرت	

استراتژی‌های گذر از خطا

استراتژی سطح عبارت

استراتژی سطح عبارت
Phrase-Level Strategy

تجزیه‌گر در هنگام کشف خطا،
یک تصحیح محلی بر روی باقیمانده‌ی ورودی انجام می‌دهد.

مانند تعویض پیشوندی از باقیمانده‌ی ورودی با رشته‌ای که به تجزیه‌گر امکان ادامه‌ی کار بدهد.
مثلاً:

- تعویض , با ;
- حذف یک ; اضافی
- درج یک ; جاافتاده

در انتخاب تعویض‌ها باید
مراقب گیرافتادن در حلقه‌های نامتناهی باشیم.

استراتژی‌های گذر از خطا

استراتژی قواعد خطا

استراتژی قواعد خطا
Error Production Strategy

قواعد گرامری خطاهای متداول به گرامر اصلی افزوده می‌شود.

اگر ایده‌های خوبی از خطاهای رایج داشته باشیم، می‌توان گرامر زبان را به‌گونه‌ای گسترش داد که رشته‌های دارای ساختارهای خطا را هم شامل شود.

سپس از این گرامر گسترش‌یافته با قواعد خطا برای ساخت تجزیه‌گر استفاده می‌کنیم.

اگر یک قاعده‌ی خطا توسط تجزیه‌گر به‌کار رفت، مانند کشف خطا با آن برخورد می‌شود و روال برخورد با خطای متناظر با آن را فراخوانی می‌کنیم و ساختار دارای خطا را گزارش می‌دهیم.

مثلاً:

- قاعده برای ساختار **if** با **then** جاافتاده

استراتژی‌های گذر از خطا

استراتژی تصحیح سراسری

استراتژی تصحیح سراسری
Global Correction Strategy

تجزیه‌گر در هنگام کشف خطا،
رشته‌ی دارای خطا را به نزدیک‌ترین رشته‌ی بدون خطا تبدیل می‌کند.

مطلوب است که کامپایلر هنگام پردازش رشته‌های ناصحیح ورودی، کمترین میزان تغییرات را اعمال کند.

رشته‌ی ورودی ناصحیح x و گرامر G در اختیار ماست:
باید یک درخت تجزیه برای رشته‌ی صحیح متناظر y پیدا شود به طوری که تعداد درج‌ها، حذف‌ها و تغییرات توکن‌های مورد نیاز برای تبدیل x به y حداقل باشد.

الگوریتم‌های می‌نیمم فاصله‌ی ویرایشی
Minimum Edit Distance

مزایا	معایب
قابل استفاده به عنوان معیاری برای ارزیابی تکنیک‌های گذر از خطا	ممکن است رشته‌ی صحیح y مورد نظر برنامه‌نویس نبوده باشد!
	هزینه‌ی زمانی و فضایی آن در پیاده‌سازی بالاست.
	بیشتر جنبه‌ی تئوری دارد.

تحلیل نحوی
برخورد با خطاهای نحوی در تجزیه‌گر

۳

برخورد با خطاهای نحوی در روش LL(1)

خاصیت «پیشوند نموپذیر»

و استفاده از آن در کشف خطا

VIABLE-PREFIX

روش‌های LL و LR دارای خاصیت «پیشوند نموپذیر» هستند

هرگاه پیشوندی از ورودی، پیشوندی از هیچ رشته‌ی معتبری در زبان نباشد، خطا کشف می‌شود

امکان کشف خطا در سریع‌ترین زمان ممکن در روش‌های LL و LR

وقوع خطا در تجزیه‌ی LL(1)

وقوع خطا در تجزیه‌ی LL(1)

۱	خالی بودن خانه‌ی $M[A,a]$ در جدول تجزیه: A (ناپایانه‌ی بالای پشته) و a (توکن جاری)
۲	عدم تطبیق پایانه‌ی بالای پشته با توکن جاری

گذر از خطا در تجزیه‌ی LL(1)

استراتژی حالت وحشت

استراتژی حالت وحشت
Panic Mode Strategy

تجزیه‌گر در هنگام کشف خطا،
نمادهای ورودی را تا رسیدن به **یک توکن مناسب** دور می‌ریزد.

توکنی که در مجموعه‌ی $\text{SYNCH}(A)$ قرار دارد
(A ناپایانه‌ی بالای پشته است).

A (ناپایانه‌ی بالای پشته)
a (توکن جاری)
b (پایانه‌ی بالای پشته)

$M[A,a] = \{\}$	خالی بودن خانه‌ی $M[A,a]$ در جدول تجزیه
	حذف **توکن جاری** a **از ورودی** و ادامه‌ی تجزیه
$M[A,a] = \text{synch}$	قرار داشتن علامت synch در خانه‌ی $M[A,a]$ در جدول تجزیه
	اگر A تنها ناپایانه‌ی موجود در پشته نباشد: حذف **ناپایانه‌ی بالای پشته** A و ادامه‌ی تجزیه
	اگر A تنها ناپایانه‌ی موجود در پشته باشد: حذف **توکن جاری** a **از ورودی** و ادامه‌ی تجزیه
$a \neq b$	عدم تطبیق **پایانه‌ی بالای پشته** با **توکن جاری**
	حذف **پایانه‌ی بالای پشته** b و ادامه‌ی تجزیه

گذر از خطا در تجزیه‌ی LL(1)

استراتژی حالت وحشت: مجموعه‌ی $\text{SYNCH}(A)$

استراتژی حالت وحشت
Panic Mode Strategy

تجزیه‌گر در هنگام کشف خطا،
نمادهای ورودی را تا رسیدن به **یک توکن مناسب** دور می‌ریزد.

توکنی که در مجموعه‌ی $\text{SYNCH}(A)$ قرار دارد
(A ناپایانه‌ی بالای پشته است).

عناصر پیشنهادی (پایانه) برای مجموعه‌ی $\text{SYNCH}(A)$

$Follow(A)$	$First(A)$	توکن‌های آغازگر ساختارها
وقتی A بالای پشته است، عمل حذف می‌تواند آن قدر انجام شود تا به نماد بعد از ساختار A برسیم.	وقتی A بالای پشته است، عمل حذف می‌تواند با کنارگذاشتن توکن‌های اضافی به نماد ابتدای ساختار A برسد.	مانند کلمات کلیدی؛ ساختارهای مختلف با این توکن‌ها شروع می‌شوند.

در خانه‌های خالی جدول تجزیه، زیر عناصر $\text{SYNCH}(A)$ نماد synch قرار می‌گیرد.

هدف از مجموعه‌ی $\text{SYNCH}(A)$ این است که تا جای ممکن توکن کمتری از ورودی حذف شود.

گذر از خطا در تجزیه‌ی LL(1)

استراتژی حالت وحشت: مثال (۱ از ۲)

$$
\begin{aligned}
E &\rightarrow TE' \\
E' &\rightarrow +TE' \mid \epsilon \\
T &\rightarrow FT' \\
T' &\rightarrow *FT' \mid \epsilon \\
F &\rightarrow \mathbf{id} \mid (E)
\end{aligned}
$$

	id	$+$	$*$	$($	$)$	$\$$
E	TE'			TE'	synch	synch
E'		$+TE'$			ϵ	ϵ
T	FT'	synch		FT'	synch	synch
T'		ϵ	$*FT'$		ϵ	ϵ
F	**id**	synch	synch	(E)	synch	synch

برای هر ناپایانه‌ی A، مجموعه‌ی $\textsc{Synch}(A)$ بر اساس $Follow$ی ناپایانه‌ی A محاسبه شده است.

گذر از خطا در تجزیه‌ی LL(1)

استراتژی حالت وحشت: مثال (۲ از ۲)

Stack	Input	Output
$\$E$	) **id** $*+$**id**$\$	error, skip)
$\$E$	**id** $*+$**id**$\$	$E \rightarrow TE'$
$\$E'T$	**id** $*+$**id**$\$	
$\$E'T'F$	**id** $*+$**id**$\$	
$\$E'T'$**id**	**id** $*+$**id**$\$	
$\$E'T'$	$*+$ **id**$\$	
$\$E'T'F*$	$*+$ **id**$\$	
$\$E'T'F$	$+$**id**$\$	error, M[F,+] = synch
$\$E'T'$	$+$**id**$\$	
$\$E'$	$+$**id**$\$	
$\$E'T+$	$+$**id**$\$	
$\$E'T$	**id**$\$	
$\$E'T'F$	**id**$\$	
$\$E'T'$**id**	**id**$\$	
$\$E'T'$	$\$	
$\$E'$	$\$	
$\$	$\$	finish

$$
\begin{aligned}
E &\rightarrow TE' \\
E' &\rightarrow +TE' \mid \epsilon \\
T &\rightarrow FT' \\
T' &\rightarrow *FT' \mid \epsilon \\
F &\rightarrow \mathbf{id} \mid (E)
\end{aligned}
$$

	id	+	*	(	)	$
E	TE'			TE'	synch	synch
E'		$+TE'$			ϵ	ϵ
T	FT'	synch		FT'	synch	synch
T'		ϵ	$*FT'$		ϵ	ϵ
F	**id**	synch	synch	(E)	synch	synch

گذر از خطا در تجزیه‌ی LL(1)

استراتژی سطح عبارت

استراتژی سطح عبارت
Phrase-Level Strategy

تجزیه‌گر در هنگام کشف خطا،
یک تصحیح محلی بر روی باقیمانده‌ی ورودی انجام می‌دهد.

انجام تصحیح محلی با
فراخوانی روال‌های اداره‌کننده‌ی خطا

در خانه‌های خالی جدول تجزیه، اشاره‌گرهایی به **روال‌های اداره‌کننده‌ی خطا** قرار داده می‌شود.
هرگاه به این خانه‌ها مراجعه شد، روال متناظر فراخوانی شده و گذر از خطا را به طور مناسب انجام می‌دهد.

عملیات روال‌های اداره‌کننده‌ی خطا	(۱)	(۲)	(۳)	(۴)	(۵)
	تغییر توکن ورودی	حذف توکن ورودی	درج توکن ورودی	حذف از بالای پشته	چاپ پیغام خطا

بایستی اطمینان پیدا کرد که در اثر تصحیح خطا امکان بروز حلقه‌های نامتناهی وجود ندارد.
(مثلاً با اطمینان از کاهش یافتن تدریجی طول باقیمانده‌ی ورودی)

گذر از خطا در تجزیه‌ی LL(1)

استراتژی قواعد خطا

استراتژی قواعد خطا

Error Production Strategy

قواعد گرامری خطاهای متداول به گرامر اصلی افزوده می‌شود.

استفاده از گرامر گسترش‌یافته با قواعد خطا برای ساخت تجزیه‌گر

گذر از خطا در تجزیه‌ی LL(1)

استراتژی قواعد خطا: مثال

استراتژی قواعد خطا
Error Production Strategy

قواعد گرامری خطاهای متداول به گرامر اصلی افزوده می‌شود.

استفاده از گرامر گسترش‌یافته با قواعد خطا برای ساخت تجزیه‌گر

S	$\rightarrow$	$S\$$	
S	$\rightarrow$	**if e then** S	
S	$\rightarrow$	**if e then** S **else** S	
S	$\rightarrow$	**a** := **e**; S	
S	$\rightarrow$	**while e do** S	

S	$\rightarrow$	$S\$$	(0)
S	$\rightarrow$	**if e then** S	(1)
S	$\rightarrow$	**if e then** S **else** S	(2)
S	$\rightarrow$	**a** := **e**; S	(3)
S	$\rightarrow$	**while e do** S	(4)
S	$\rightarrow$	**a** := **e** S	(-3)
S	$\rightarrow$	**while e** S	(-4)

قاعده‌ی (-3): متداول بودن خطای جا افتادن ;
قاعده‌ی (-4): متداول بودن خطای جا افتادن **do**

تحلیل نحوی
برخورد با خطاهای نحوی در تجزیه‌گر

۴

برخورد با خطاهای نحوی در روش‌های تقدم

وقوع خطا در روش‌های تجزیه‌ی تقدم

وقوع خطا در روش‌های تجزیه‌ی تقدم

۱	عدم وجود رابطه میان X (نماد بالای پشته) و b (توکن جاری)
۲	عدم تطبیق **دستگیره‌ی واقع در بالای پشته با سمت راست هیچ یک از قواعد**

گذر از خطا در روش‌های تجزیه‌ی تقدم

استراتژی سطح عبارت (برای حالت عدم وجود رابطه)

استراتژی سطح عبارت
Phrase-Level Strategy

تجزیه‌گر در هنگام کشف خطا،
یک تصحیح محلی بر روی باقیمانده‌ی ورودی انجام می‌دهد.

انجام تصحیح محلی با
فراخوانی روال‌های اداره‌کننده‌ی خطا

در خانه‌های خالی جدول تجزیه، اشاره‌گرهایی به **روال‌های اداره‌کننده‌ی خطا** قرار داده می‌شود.
هرگاه به این خانه‌ها مراجعه شد، روال متناظر فراخوانی شده و گذر از خطا را به طور مناسب انجام می‌دهد.

عملیات روال‌های اداره‌کننده‌ی خطا	(۱)	(۲)	(۳)	(۴)	(۵)
	تغییر توکن ورودی	حذف توکن ورودی	درج توکن ورودی	حذف از بالای پشته	چاپ پیغام خطا

بایستی اطمینان پیدا کرد که در اثر تصحیح خطا امکان بروز حلقه‌های نامتناهی وجود ندارد.
(مثلاً با اطمینان از کاهش یافتن تدریجی طول باقیمانده‌ی ورودی)

گذر از خطا در روش‌های تجزیه‌ی تقدم

استراتژی سطح عبارت (برای حالت عدم وجود رابطه): مثال (۱ از ۲)

$$E \rightarrow E + F \mid F$$
$$F \rightarrow (E) \mid \mathbf{id}$$

	$+$	$($	$)$	**id**	$\$$
$+$	$\gtrdot$	$\lessdot$	$\gtrdot$	$\lessdot$	$\gtrdot$
$($	$\lessdot$	$\lessdot$	$\doteq$	$\lessdot$	e_3
$)$	$\gtrdot$	e_2	$\gtrdot$	e_2	$\gtrdot$
id	$\gtrdot$	e_2	$\gtrdot$	e_2	$\gtrdot$
$\$$	$\lessdot$	$\lessdot$	e_1	$\lessdot$	$\doteq$

e_1	عبارت با پرانتز بسته شروع شده است
e_2	پس از **id** یا پرانتز بسته، **id** یا پرانتز باز آمده است (گم شدن عملگر بین آنها)
e_3	عبارت با پرانتز باز خاتمه یافته است

گذر از خطا در روش‌های تجزیه‌ی تقدم

استراتژی سطح عبارت (برای حالت عدم وجود رابطه): مثال (۲ از ۲)

$$E \rightarrow E + F \mid F$$
$$F \rightarrow (E) \mid \mathbf{id}$$

	$+$	$($	$)$	**id**	$\$$
$+$	$\gtrdot$	$\lessdot$	$\gtrdot$	$\lessdot$	$\gtrdot$
$($	$\lessdot$	$\lessdot$	$\doteq$	$\lessdot$	e_3
$)$	$\gtrdot$	e_2	$\gtrdot$	e_2	$\gtrdot$
id	$\gtrdot$	e_2	$\gtrdot$	e_2	$\gtrdot$
$\$$	$\lessdot$	$\lessdot$	e_1	$\lessdot$	$\doteq$

e_1	توکن (را حذف کن	پیام خطا: unbalanced parenthesis
e_2	توکن + را به ورودی اضافه کن	پیام خطا: missing operator
e_3	توکن) را از بالای پشته حذف کن	پیام خطا: missing parenthesis

گذر از خطا در روش‌های تجزیه‌ی تقدم

استراتژی سطح عبارت (برای حالت عدم تطابق با سمت راست قاعده)

استراتژی سطح عبارت
Phrase-Level Strategy

تجزیه‌گر در هنگام کشف خطا،
یک تصحیح محلی بر روی باقیمانده‌ی ورودی انجام می‌دهد.

انجام تصحیح محلی با
فراخوانی روال‌های اداره‌کننده‌ی خطا

در خانه‌های خالی جدول تجزیه، اشاره‌گرهایی به **روال‌های اداره‌کننده‌ی خطا** قرار داده می‌شود.
هرگاه به این خانه‌ها مراجعه شد، روال متناظر فراخوانی شده و گذر از خطا را به طور مناسب انجام می‌دهد.

عملیات روال‌های اداره‌کننده‌ی خطا

- تجزیه‌گر به دنبال قاعده‌ای می‌گردد که سمت راست آن بیشترین شباهت را به دستگیره داشته باشد.
- با توجه به اختلاف دستگیره با سمت راست این قاعده،
  ۱) پیام خطای مناسبی چاپ می‌شود، ۲) کاهش انجام می‌شود، ۳) تجزیه ادامه می‌یابد.

گذر از خطا در روش‌های تجزیه‌ی تقدم

استراتژی سطح عبارت (برای حالت عدم تطابق با سمت راست قاعده): مثال

دستگیره	$aEbc$
قاعده‌ی یافت شده	$E \rightarrow aEc$
خطا: b اضافی در ورودی	

دستگیره	$abEc$
قاعده‌ی یافت شده	$E \rightarrow abEdc$
خطا: d گم‌شده در ورودی	

دستگیره	abc
قاعده‌ی یافت شده	$E \rightarrow aEbc$
خطا: ساختار نحوی E مورد انتظار است (مثلاً یک عبارت ریاضی جا افتاده)	

* در این حالت، پیام خطا نباید به طور مستقیم به ناپایانه‌ی E اشاره کند،
زیرا کاربر در مورد ناپایانه‌های گرامر چیزی نمی‌داند!

اصول طراحی کامپایلر

تحلیل نحوی
برخورد با خطاهای نحوی در تجزیه‌گر

۵

برخورد با خطاهای نحوی در روش‌های LR

وقوع خطا در روش‌های تجزیه‌ی LR

وقوع خطا در روش‌های تجزیه‌ی LR

- مراجعه به یک خانه‌ی خالی از بخش ACTION جدول تجزیه

گذر از خطا در روش‌های تجزیه‌ی LR

استراتژی حالت وحشت

استراتژی حالت وحشت
Panic Mode Strategy

تجزیه‌گر در هنگام کشف خطا،
نمادهای ورودی را تا رسیدن به **یک توکن مناسب** دور می‌ریزد.

توکنی که در مجموعه‌ی $Follow(A)$ قرار دارد
(A ناپایانه‌ی بالای پشته است).

در صورت برخورد با خطای نحوی
(خانه‌ی خالی)

- تجزیه‌گر در پشته پایین می‌رود تا به حالتی برسد که حداقل یک مدخل GOTO داشته باشد.
  (حالت s برای ناپایانه‌ی A در خانه‌ی $\text{GOTO}[s,A]$ مقداری مانند n دارد)
- A و n به ترتیب به بالای پشته اضافه می‌شوند.
- از توکن‌های ورودی آن قدر حذف می‌کند تا به یکی از عناصر $Follow(A)$ رسد.

پیاده‌سازی این استراتژی بسیار ساده است، اما ممکن است بخش زیادی از ورودی حذف شود!

گذر از خطا در روش‌های تجزیه‌ی LR

استراتژی سطح عبارت

استراتژی سطح عبارت
Phrase-Level Strategy

تجزیه‌گر در هنگام کشف خطا،
یک تصحیح محلی بر روی باقیمانده‌ی ورودی انجام می‌دهد.

انجام تصحیح محلی با
فراخوانی روال‌های اداره‌کننده‌ی خطا

در خانه‌های خالی بخش ACTION جدول تجزیه، اشاره‌گرهایی به **روال‌های اداره‌کننده‌ی خطا** قرار داده می‌شود.
هرگاه به این خانه‌ها مراجعه شد، روال متناظر فراخوانی شده و گذر از خطا را به طور مناسب انجام می‌دهد.

در روال‌های اداره‌کننده‌ی خطا

- هیچ بخشی از ورودی، بدون بررسی حذف نمی‌شود.
- با استراتژی سطح عبارت، خطا در همان محل وقوع، تصحیح می‌شود.

قرار دادن روال‌ها در جدول تجزیه بخش ACTION

در حالت‌هایی که **کاهش فقط با یک قاعده‌ی خاص** انجام می‌شود،
خانه‌های خالی را با عمل کاهش با همان قاعده پر می‌کنیم.
(این کار باعث می‌شود کشف خطا پس از چند مرحله کاهش و پیش از اولین عمل شیفت انجام شود.)
سایر عناصر خالی جدول تجزیه، با اشاره‌گرهایی به روال‌های گذر از خطا پر می‌شود.

گذر از خطا در روش‌های تجزیه‌ی LR

استراتژی سطح عبارت: مثال (۱ از ۵)

$$E \rightarrow E + E \mid E * E \mid (E) \mid \mathbf{id} \quad (1, 2, 3, 4)$$

	ACTION						GOTO
STATE	**id**	+	*	(	)	$	E
0	s3			s2			1
1		s4	s5			acc	
2	s3			s2			6
3		r4	r4		r4	r4	
4	s3			s2			7
5	s3			s2			8
6		s4	s5		s9		
7		r1	s5		r1	r1	
8		r2	r2		r2	r2	
9		r3	r3		r3	r3	

گذر از خطا در روش‌های تجزیه‌ی LR

استراتژی سطح عبارت: مثال (۲ از ۵)

$$E \rightarrow E + E \mid E * E \mid (E) \mid \mathbf{id} \quad (1, 2, 3, 4)$$

	ACTION						GOTO
STATE	**id**	+	*	(	)	$	E
0	s3			s2			1
1		s4	s5			acc	
2	s3			s2			6
3		r4	r4		r4	r4	
4	s3			s2			7
5	s3			s2			8
6		s4	s5		s9		
7		r1	s5		r1	r1	
8		r2	r2		r2	r2	
9		r3	r3		r3	r3	

در حالت‌هایی که **کاهش فقط با یک قاعده‌ی خاص** انجام می‌شود،
خانه‌های خالی را با عمل کاهش با همان قاعده پر می‌کنیم.

گذر از خطا در روش‌های تجزیه‌ی LR

استراتژی سطح عبارت: مثال (۳ از ۵)

$$E \to E + E \mid E * E \mid (E) \mid \mathbf{id} \quad (1, 2, 3, 4)$$

	ACTION						GOTO
STATE	**id**	+	*	(	)	$	E
0	s3	e_1	e_1	s2	e_2	e_1	1
1	e_3	s4	s5	e_3	e_2	acc	
2	s3	e_1	e_1	s2	e_2	e_1	6
3	r4	r4	r4	r4	r4	r4	
4	s3	e_1	e_1	s2	e_2	e_1	7
5	s3	e_1	e_1	s2	e_2	e_1	8
6	e_3	s4	s5	e_3	s9	e_4	
7	r1	r1	s5	r1	r1	r1	
8	r2	r2	r2	r2	r2	r2	
9	r3	r3	r3	r3	r3	r3	

e_1	در حالت‌های 0، 2، 4، 5 در صورت انتظار داشتن **id** یا)، اما یافتن یک عملگر یا انتهای ورودی
e_2	در حالت‌های 0، 1، 2، 4، 5 در هنگام مشاهده‌ی یک پرانتز بسته اضافی
e_3	در حالت‌های 1 و 6 در صورت انتظار داشتن یک عملگر، اما یافتن **id** یا (
e_4	در حالت 6 به انتهای ورودی برسیم، در صورت انتظار داشتن یک عملگر یا (

گذر از خطا در روش‌های تجزیه‌ی LR

استراتژی سطح عبارت: مثال (۴ از ۵)

$$E \to E + E \mid E * E \mid (E) \mid \mathbf{id} \quad (1, 2, 3, 4)$$

	Action						Goto
State	**id**	+	*	(	)	$	E
0	s3	e_1	e_1	s2	e_2	e_1	1
1	e_3	s4	s5	e_3	e_2	acc	
2	s3	e_1	e_1	s2	e_2	e_1	6
3	r4	r4	r4	r4	r4	r4	
4	s3	e_1	e_1	s2	e_2	e_1	7
5	s3	e_1	e_1	s2	e_2	e_1	8
6	e_3	s4	s5	e_3	s9	e_4	
7	r1	r1	s5	r1	r1	r1	
8	r2	r2	r2	r2	r2	r2	
9	r3	r3	r3	r3	r3	r3	

e_1	قرار دادن **id** و سپس حالت $\text{GOTO}(0; 2; 4; 5, \mathbf{id}) = 3$ در پشته	پیام خطا: missing operand
e_2	حذف) از ورودی	پیام خطا: unbalanced right parenthesis
e_3	قرار دادن + و سپس حالت $\text{GOTO}(1; 6, +) = 4$ در پشته	پیام خطا: missing operator
e_4	قرار دادن (و سپس حالت $\text{GOTO}(6, () = 9$ در پشته	پیام خطا: missing right parenthesis

Prepared by Kazim Fouladi | Fall 2017 | 3rd Edition
کاظم فولادی

گذر از خطا در روش‌های تجزیه‌ی LR

استراتژی سطح عبارت: مثال (۵ از ۵)

Stack	Input	Action	Message
0	**id**+)$		
0 **id** 3	+)$		
0 E 1	+)$		
0 E 1 + 4	)$	error e_2: remove (	unbalanced right parenthesis
0 E 1 + 4	$	error e_1: push **id** 3 on stack	missing operand
0 E 1 + 4 **id** 3	$		
0 E 1 + 4 E 7	$		
0 E 1	$	finish	

گذر از خطا در روش‌های تجزیه‌ی LR

استراتژی سطح عبارت: مثال (۵ از ۵)

Stack	Input	Action	Message
0	**id**+)$		
0 **id** 3	+)$		
0 E 1	+)$		
0 E 1 + 4	)$	error e_2: remove (	unbalanced right parenthesis
0 E 1 + 4	$	error e_1: push **id** 3 on stack	missing operand
0 E 1 + 4 **id** 3	$		
0 E 1 + 4 E 7	$		
0 E 1	$	finish	

$E \rightarrow E + E \mid E * E \mid (E) \mid \mathbf{id} \quad (1, 2, 3, 4)$

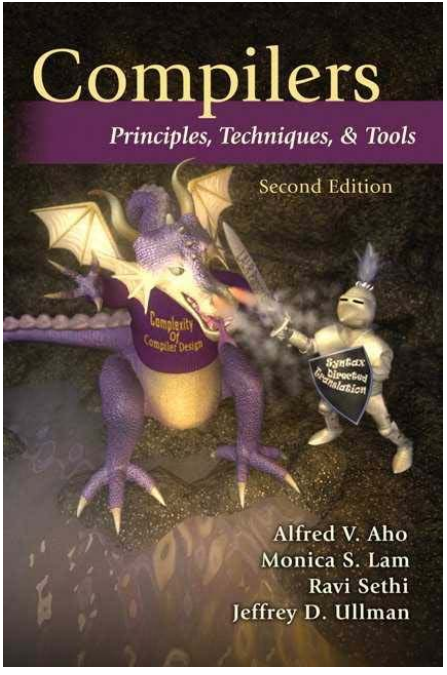
	Action						Goto
State	**id**	+	*	(	)	$	E
0	s3	e_1	e_1	s2	e_2	e_1	1
1	e_3	s4	s5	e_3	e_2	acc	
2	s3	e_1	e_1	s2	e_2	e_1	6
3	r4	r4	r4	r4	r4	r4	
4	s3	e_1	e_1	s2	e_2	e_1	7
5	s3	e_1	e_1	s2	e_2	e_1	8
6	e_3	s4	s5	e_3	s9	e_4	
7	r1	r1	s5	r1	r1	r1	
8	r2	r2	r2	r2	r2	r2	
9	r3	r3	r3	r3	r3	r3	

تحلیل نحوی
برخورد با خطاهای نحوی در تجزیه‌گر

۶

منابع

منبع اصلی



A. V. Aho, M. S. Lam, R. Sethi, J. D. Ullman,
Compilers: Principles, Techniques and Tools,
Second Edition, Addison-Wesley, 2007.

Chapter 4 (4.1, 4.4.5, 4.8.3)