



دانشگاه صنعتی اصفهان
دانشکده مهندسی برق و کامپیوتر

عنوان: تکلیف تئوری دوم درس کامپایلر

نام و نام خانوادگی: علیرضا ابره فروش

شماره دانشجویی: ۹۸۱۶۶۰۳

نیم سال تحصیلی: بهار ۱۴۰۲/۱۴۰۱

مدرس: دکتر حسین فلسفین

۱

a ۱.۱

طبق الگوریتم زیر عمل می‌کنیم.

Algorithm 4.21: Left factoring a grammar.**INPUT:** Grammar G .**OUTPUT:** An equivalent left-factored grammar.**METHOD:** For each nonterminal A , find the longest prefix α common to two or more of its alternatives. If $\alpha \neq \epsilon$ — i.e., there is a nontrivial common prefix — replace all of the A -productions $A \rightarrow \alpha\beta_1 \mid \alpha\beta_2 \mid \cdots \mid \alpha\beta_n \mid \gamma$, where γ represents all alternatives that do not begin with α , by

$$\begin{aligned} A &\rightarrow \alpha A' \mid \gamma \\ A' &\rightarrow \beta_1 \mid \beta_2 \mid \cdots \mid \beta_n \end{aligned}$$

Here A' is a new nonterminal. Repeatedly apply this transformation until no two alternatives for a nonterminal have a common prefix. \square

شکل ۱

$$\begin{aligned} S &\rightarrow SS' \\ S' &\rightarrow +S \mid +P \\ P &\rightarrow PP' \\ P' &\rightarrow *P \mid *I \\ I &\rightarrow -I \mid (S) \mid D \\ D &\rightarrow 0 \mid 1N \\ N &\rightarrow NN' \mid 0 \mid 1 \mid \varepsilon \\ N' &\rightarrow N \end{aligned}$$

b ۲.۱

طبق الگوریتم زیر عمل می‌کنیم.

Immediate left recursion can be eliminated by the following technique, which works for any number of A -productions. First, group the productions as

$$A \rightarrow A\alpha_1 | A\alpha_2 | \dots | A\alpha_m | \beta_1 | \beta_2 | \dots | \beta_n$$

where no β_i begins with an A . Then, replace the A -productions by

$$\begin{aligned} A &\rightarrow \beta_1 A' | \beta_2 A' | \dots | \beta_n A' \\ A' &\rightarrow \alpha_1 A' | \alpha_2 A' | \dots | \alpha_m A' | \epsilon \end{aligned}$$

The nonterminal A generates the same strings as before but is no longer left recursive.

$$\begin{aligned} S &\longrightarrow US' \\ S' &\longrightarrow aSS' | \epsilon \\ U &\longrightarrow TU' \\ U' &\longrightarrow uUU' | \epsilon \\ T &\longrightarrow tT' | fT' | (S)T' \\ T' &\longrightarrow |nT' | \epsilon \end{aligned}$$

۲

۱.۲

خطاهای شناسایی شده توسط تحلیلگر لغوی معمولاً دارای ویژگی‌های زیر هستند:

- این خطاها زمانی رخ می‌دهند که دنباله ورودی از کاراکترهای معتبر در زبان برنامه‌نویسی مورد نظر تشخیص داده نمی‌شود.
- این خطاها به طور کلی توسط تحلیلگر لغوی تشخیص داده می‌شوند که در مرحله اول فرایند کامپایل است.
- این خطاها اغلب به دلیل خطاهای نحوی، مانند کلمات کلیدی یا شناسه‌های نوشتاری نادرست یا کاراکترهای نامعتبر یا نمادها ایجاد می‌شوند.

۲.۲

چهار نمونه مختلف از انواع خطاهای شناسایی شده توسط تحلیلگر لغوی، شامل موارد زیر می‌شوند:

۱. کاراکترهای غیرمجاز: این کاراکترها که در زبان برنامه‌نویسی شناخته نشده‌اند، مانند کاراکترهای غیر چاپ‌پذیر یا نویسه‌هایی از زبان‌های دیگر هستند.

۲. عدم تطابق نقل قول: این خطا هنگامی رخ می‌دهد که یک متن رشته‌ای به درستی با نقل قول متناظر خاتمه نمی‌یابد. به عنوان مثال، "Hello World!" احتمالاً به دلیل عدم وجود دابل کوتیشن پایانی، با خطای لغوی روبرو می‌شود.

۳.

۳

کامپایلر برنامه‌ای است که می‌تواند یک برنامه از یک زبان (زبان منبع) را بخواند و آن را به یک برنامه معادل به زبان دیگری (زبان هدف) ترجمه کند. نقش مهمی که کامپایلر ایفا می‌کند، گزارش هر نوع خطایی در برنامه‌ی منبع است که در طول فرآیند ترجمه تشخیص داده می‌شود.

منابع

□□□□□□□□□□