

## اصول طراحی کامپایلر - تمرینات سری دوم - خلیلیان

۱. یک ماشین خودکار قطعی (dfa) برای کلیه اعداد مبنای دو رسم کنید که تعداد صفرهای آن زوج و تعداد یک‌های آن نیز زوج باشد. در ضمن دست‌کم دو صفر و دو عدد یک در این اعداد موجود باشند.
۲. یک ماشین خودکار قطعی بهینه برای عبارت منظم  $(albc)^* abb$  ایجاد نمایید و سپس برنامه تحلیلگر لغوی را برای تشخیص این‌گونه رشته‌ها به زبان جاوا یا سی‌شارپ بنویسید. ابتدا عبارات را به ماشین غیرقطعی، سپس به ماشین قطعی و نهایتاً به ماشین قطعی بهینه تبدیل کنید.
۳. یک عبارت منظم برای کلیه رشته‌های اعداد مبنای دو بنویسید که تعداد صفرها در آنها فرد و تعداد یک‌ها زوج است. سپس آنرا به ماشین خودکار قطعی بهینه (کمینه) تبدیل کنید.
۴. برای سهولت در نمادگذاری، می‌توانیم برای عبارتهای منظم نام‌هایی در نظر بگیریم و آنها را تعریف کنیم. تعریف منظم دنباله‌ای به صورت زیر است:

$d_1 \rightarrow r_1$   
 $d_2 \rightarrow r_2$   
 .....  
 $d_n \rightarrow r_n$

که در  $d_i$  یک نام متمایز است و هر  $r_i$  یک عبارت منظم. با جایگذاری مکرر  $d_i$ ها با  $r_i$ ها، می‌توانیم روی الفبای مورد نظر خود، عبارات منظم بسازیم. در ذیل تعاریف منظم برای شناسه‌ها و اعداد بدون علامت در پاسکال ارائه شده است. برای این تعاریف منظم ماشین خودکار رسم نموده و آنها را به ماشین خودکار قطعی بهینه تبدیل نمایید. در صورت تمایل برنامه تحلیلگر لغوی آنها را نیز بنویسید.

$letter \rightarrow A|B|\dots|Z|a|b|\dots|z$   
 $digit \rightarrow 0|1|2|\dots|9$   
 $id \rightarrow letter(letter|digit)^*$

$digit \rightarrow 0|1|2|\dots|9$   
 $digits \rightarrow digit|digit^*$   
 $optional\_fraction \rightarrow \cdot digit|e$   
 $optional\_exponent \rightarrow (E(+|-|e)digits)|e$   
 $num \rightarrow digits \ optional\_fraction \ optional\_exponent$

توجه کنید که برای نمایش خلاصه‌تر نمادها گاهی اوقات از نماد  $letter \rightarrow [A-Z]$  به جای  $letter \rightarrow A|B|\dots|Z$  استفاده می‌شود که آنرا کلاس کاراکترها می‌نامند.

۵. به زبان فارسی بیان کنید که هر یک از عبارات منظم زیر چه زبانی را تولید می‌کنند.

$0(01)^*0$   
 $(01)^*0(01)(01)$

۶. تعاریف منظمی برای زبان‌های زیر ارائه دهید.

الف) تمام رشته‌های حرفی انگلیسی که شامل پنج حرف صدادر هستند.

ب) تمام رشته‌های حرفی که در آنها حروف به ترتیب صعودی هستند.

۷. برای عبارت منظم زیر dfa با حداقل تعداد حالت بسازید.

$(alb)^* a(albb)^*$

۸. مزیت کد میانی در کامپایلر چیست؟

۹. چرا در تحلیلگر لغوی از حافظه موقت (بافر) استفاده می‌شود؟

۱۰. مزیت استفاده از بافر دو نیمه چیست؟ چگونه می‌توان عملیات آنرا تسریع نمود؟

۱۱. توضیح دهید کدام عامل می‌تواند سرعت یک تحلیلگر لغوی را افزایش دهد و چگونه؟

۱۲. چه عاملی باعث می‌شود که کامپایلر مجبور به داشتن بیش از یک گذر گردد؟

۱۳. برای هر یک از زبان‌های زیر یک dfa رسم کنید.

$L1 = \{w \in \{a,b\}^* : n_a(w) \bmod 2 = 0\}$   
 $L2 = \{w \in \{a,b\}^* : |w| \bmod 5 \neq 0\}$   
 $L3 = \{w \in \{a,b\}^* : n_a(w) \bmod 2 > n_b(w) \bmod 2\}$   
 $L4 = \{vwv : |v|=2, v, w \in \{a,b\}^*\}$



## اصول طراحی کامپایلر - تمرینات سری دوم - خلیلیان

۱۴. یک nfa با حداکثر ۵ وضعیت طراحی کنید که زبان  $\{abab^n: n \geq 0\} \cup \{aba^n: n \geq 0\}$  را بپذیرد.

۱۵. یک nfa با چهار وضعیت برای زبان  $L = \{a^n: n \geq 0\} \cup \{b^na: n \geq 1\}$  طراحی کنید.

۱۶. برای زبان‌های زیر یک nfa رسم کنید. سپس آنرا به dfa تبدیل کرده و نهایتاً dfa را کمینه کنید.

$$L1 = \{a^n b^m: n \geq 2, m \geq 1\}$$

$$L2 = \{a^n b: n \geq 0\} \cup \{b^n a: n \geq 1\}$$

۱۷. برای زبان‌های زیر عبارات منظمی بنویسید.

$$L1 = \{a^n b^m: n \geq 1, m \geq 1, nm \geq 3\}$$

$$L2 = \{ab^n w: n \geq 3, w \in \{a, b\}^+\}$$

$$L3 = \{w: |w| \bmod 3 = 0\}$$

۱۸. برای هر یک از زبان‌های زیر روی الفبای  $\Sigma = \{a, b, c\}$  یا  $\Sigma = \{0, 1\}$  یک عبارت منظم بنویسید.

الف) تمام رشته‌هایی که دقیقاً شامل یک a هستند.

ب) تمام رشته‌هایی که حداکثر شامل ۲ رشته aa باشند.

ج) تمام رشته‌هایی که به 01 ختم می‌شوند.