



دانشکده مهندسی کامپیوتر

# اصول طراحی کامپایلر (دکتر ممتازی)

نیم سال دوم سال تحصیلی ۱۴۰۱-۱۴۰۲

## تمرین سری اول



دانشگاه صنعتی امیرکبیر  
(پلی تکنیک تهران)

قبل از حل سوالات به نکات زیر توجه فرمایید:

- هدف از انجام تمرین‌ها، یادگیری عمیق‌تر مطالب درسی است. در نتیجه هرگونه تقلب موجب کسر نمره خواهد شد.
- مهلت تحویل تا پایان ۱۹ فروردین و نحوه تحویل از طریق سامانه کورسز است. (همه پاسخ‌ها را به صورت یک فایل فشرده ارسال کنید و نام فایل را شماره دانشجویی قرار دهید.)
- در صورت وجود هرگونه سوال می‌توانید از طریق تلگرام یا ایمیل [CompilerSpringAut2023@gmail.com](mailto:CompilerSpringAut2023@gmail.com) با تدریس‌یاران در ارتباط باشید. در صورت مشکل در بارگذاری پاسخ‌ها فقط از طریق ایمیل اقدام نمایید و در موضوع ایمیل شماره تمرین، نام و نام خانوادگی و شماره دانشجویی را ذکر کنید.

۱. برای زبان‌های زیر اتوماتای متناهی غیر قطعی ارائه دهید. توضیح دهید که چرا طراحی اتوماتای غیر قطعی از قطعی ساده‌تر است. سپس برای فقط یکی از زبان‌ها، پذیرش رشته‌ای با طول حداقل ۵ را با الگوریتم شبیه سازی کارای اتوماتای متناهی غیر قطعی<sup>۱</sup> بررسی کنید. (در هر مرحله از الگوریتم مجموعه استیت‌هایی از NFA را که ممکن است در آنها قرار گرفته باشیم را مشخص کنید.) ( $\Sigma = \{a, b, c\}$ )

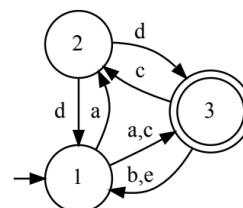
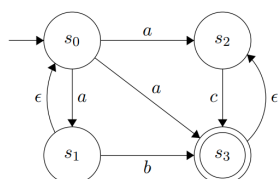
(آ) زبان  $L_1$  شامل کلماتی باشد که حداکثر یکی از حروف الفبا در آن بیش از یک بار تکرار شده‌اند.

$\epsilon \in L_1, a \in L_1, abb \in L_1, abbc \in L_1, abcab \notin L_1, aabb \notin L_1$

(ب) زبان  $L_2$  شامل کلماتی به فرم  $sx^n$  است که  $|x| = 1, n \geq 1$  و  $s \in \Sigma^*$  در  $s$  ظاهر نشده است.

$ababc \in L_2, bbb \in L_2, aaacbb \in L_2, \epsilon \notin L_2, baaab \notin L_2, abcabc \notin L_2$

۲. اتوماتاهای متناهی غیرقطعی زیر را با الگوریتم مطرح شده در کلاس به اتوماتاهای قطعی تبدیل کنید.



۳. برای زبان‌های زیر اتوماتای متناهی قطعی ارائه دهید. برای هر استیت توصیف کوتاهی از کاربرد آن (یا توصیف کوتاهی از مجموعه رشته‌هایی که در آن قرار دارند) ارائه دهید. ( $\Sigma = \{a, b\}$ )

(a)  $L_1 = \{x \in \Sigma^* \mid \text{there are odd number of } a\text{'s in } x \text{ or (logical or) } aba \text{ is a substring of } x\}$ .

<sup>1</sup>The algorithm for Efficient Simulation of a string on NFA runs in  $\mathcal{O}(mn^2)$  where  $m$  is the length of string and  $n$  is the number of states of the NFA while the slow method is to first convert the NFA to a DFA using the subset construction method, which in turn runs in exponential time with respect to the number of the states of the NFA, and then simulate the resulting DFA on the given string in linear time. (Read section 3.7.2 from *Compilers: Principles, Techniques, and Tools*, 2nd edition, Aho et al.)

(b)  $L_2 = \{x \in \Sigma^* \mid \text{there are exactly three } a\text{'s between every two consecutive } b\text{'s in } x\}$ . ( $baaabaaab \in L_2, baabaaab \notin L_2, baaaa \in L_2, baaaba \in L_2$ )

۴. برای هر یک از زبان‌های زیر، اگر زبان منظم است عبارت منظم آن را بیابید. اگر زبان منظم نیست دلیل ارائه کنید. نیازی به اثبات نیست.  
 $(\Sigma = \{a, b\})$

(آ) زبان همه رشته‌هایی که هیچ دو حرف متوالی یکسان نباشند.

(ب) زبان همه رشته‌هایی که با حروف یکسان شروع و خاتمه می‌یابند.

(ج) زبان همه رشته‌هایی که تعداد زوج  $a$  دارند.

(د) زبان همه رشته‌هایی که شامل  $bab$  نباشند.

(ه) زبان همه رشته‌هایی که به  $aab$  ختم نشوند.

(و) زبان همه رشته‌های به طول فرد که  $a$  در ابتدا، انتها و میانه رشته ظاهر شده باشد.

(ز) زبان همه رشته‌هایی که با  $aa$  شروع می‌شوند و طول آنها فرد است.

(ح) زبان همه رشته‌هایی که تعداد  $a$  ها از  $b$  ها در هر پسوند آن کمتر باشد.

۵. ابزارهای تولیدکننده تحلیل‌گر لغوی<sup>۲</sup> مانند flex, ply, jflex و ...، استراتژی‌های مختلفی برای تبدیل رشته ورودی به توکن‌ها با استفاده از تعریف‌های منظم تعریف شده توسط کاربر دارند. یکی از این استراتژی‌ها این است که عبارت منظمی برای تولید توکن استفاده می‌شود که طولانی‌ترین پیشوند از رشته فعلی با آن تطابق یابد. به عنوان مثال اگر مجموعه تعریف‌های منظم به صورت زیر باشد، رشته  $dot$  تنها یک توکن  $id$  تولید می‌کند و نه دو توکن  $do$  و  $id$ .

$$\{\langle id, [A - Za - z][A - Za - z0 - 9]^* \rangle, \langle do, do \rangle\}$$

اما مجموعه‌ای از تعریف‌های منظم وجود دارند که بتوان به کمک آنها رشته ورودی را به دنباله‌ای از توکن‌ها تبدیل کرد ولی استراتژی فوق امکان انجام این کار را نداشته باشد. مثالی از یک رشته و یک مجموعه از تعریف‌های منظم ارائه دهید که رشته قابل شکستن به تعدادی زیررشته باشد که هر کدام با یکی از عبارت‌های منظم درون مجموعه تطابق داشته باشند ولی استراتژی فوق قادر به تولید هیچ دنباله‌ای از توکن‌ها نباشد و اعلام خطا کند.

۶. (امتیازی)

(آ) عبارتی منظم برای زبانی روی الفبای  $\Sigma = \{a, b\}$  بیابید که کلمات آن شامل تعداد زوج  $a$  و زوج  $b$  باشند.

(ب) فرض کنید  $A, B, X$  زبان‌هایی روی الفبای  $\Sigma$  باشند به طوری که  $X = A \cdot X \cup B$  و  $\epsilon \notin A$ . نشان دهید  $X = A^* \cdot B$ .

در واقع این مسئله نشان می‌دهد که اگر  $X = A \cdot X \cup B$  را به نوعی یک معادله زبانی بر حسب  $X$  ببینیم، دارای جواب یکتا  $X = A^* \cdot B$  خواهد بود اگر  $\epsilon \notin A$ .

(ج) اگر  $\epsilon \in A$  و  $X = A^* \cdot B$  تنها جواب معادله نخواهد بود.  $A$  و  $B$  را بگونه‌ای ارائه دهید که  $X \neq A^* \cdot B$  یافت شود که در معادله مذکور صدق کند.

(د) از بخش اول و دوم استفاده کنید و عبارتی منظم برای زبانی روی الفبای  $\Sigma = \{a, b\}$  بیابید که کلمات آن شامل تعداد زوج  $a$  و فرد  $b$  باشند.

<sup>2</sup>Lexical Analyzer Generator