



دانشگاه صنعتی اصفهان
دانشکده مهندسی برق و کامپیوتر

عنوان: تکلیف تئوری اول درس کامپایلر

نام و نام خانوادگی: علیرضا ابره فروش

شماره دانشجویی: ۹۸۱۶۶۰۳

نیم سال تحصیلی: بهار ۱۴۰۲/۱۴۰۱

مدرس: دکتر حسین فلسفین

۱

۱.۱

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.

۲.۱

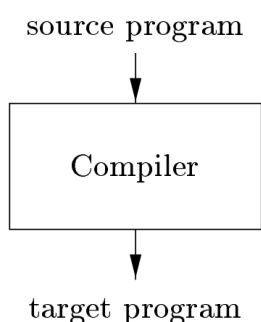
چهار نمونه مختلف از انواع خطاهای شناسایی شده توسط تحلیلگر لغوی، شامل موارد زیر می‌شوند:

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.

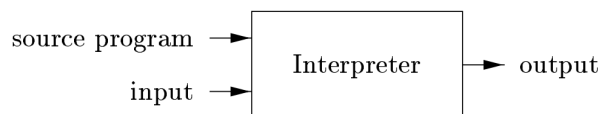
۲

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.



شکل ۱

یک مترجم (مفسر) نیز یک نوع دیگر از پردازشگر زبان است. به جای تولید یک برنامه هدف به عنوان ترجمه، یک مترجم به نظر می‌رسد که به طور مستقیم عملیات مشخص شده در برنامه‌ی منبع را در ورودی‌هایی تأمین شده توسط کاربر اجرا می‌کند.



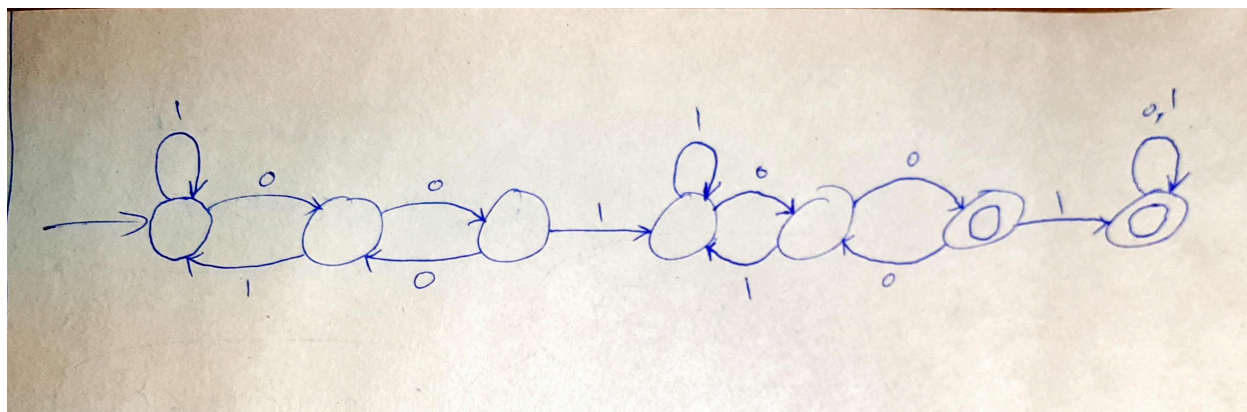
شکل ۲

۳

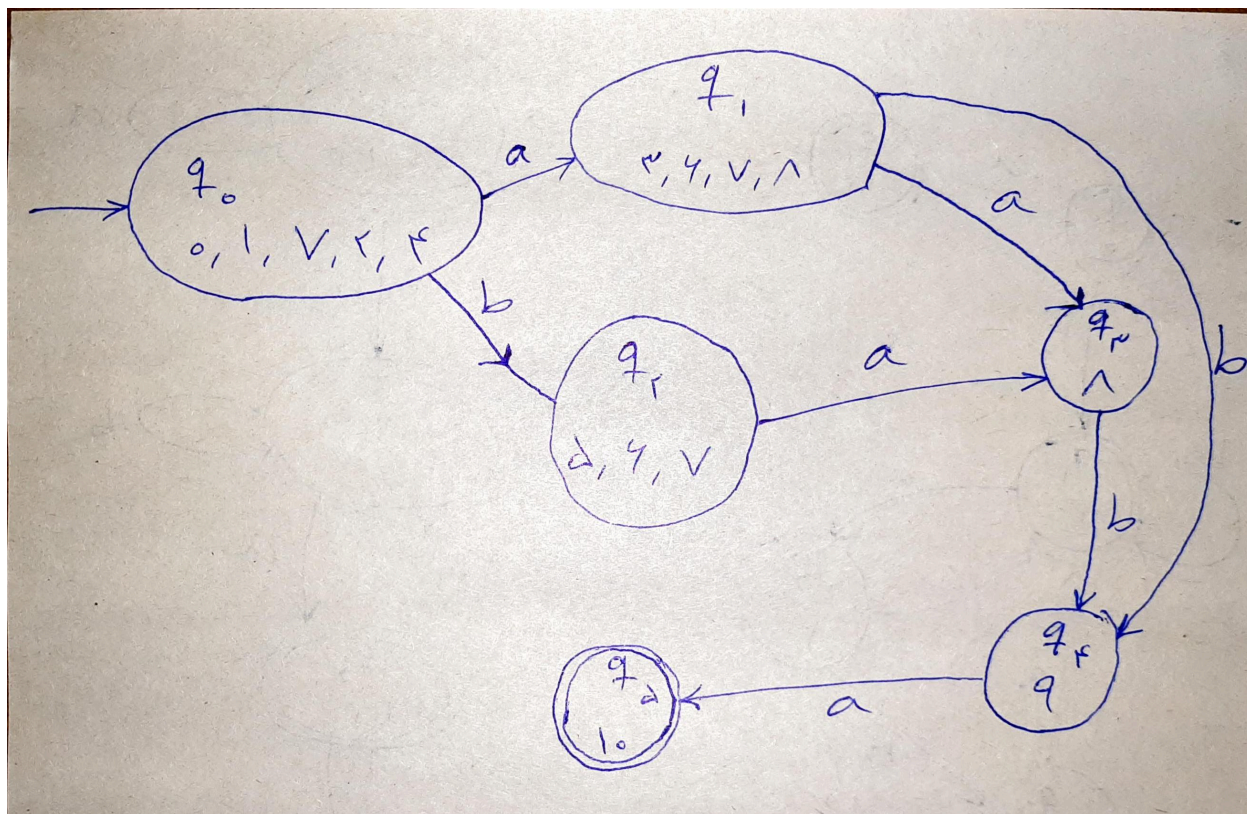
۱.۳

$$(\varepsilon \cup (0 \cup 1)^* 1) (00)^+ (1 \cup 1 (0 \cup 1)^* 1) (00)^+ (1 (0 \cup 1)^* \cup \varepsilon)$$

۲.۳

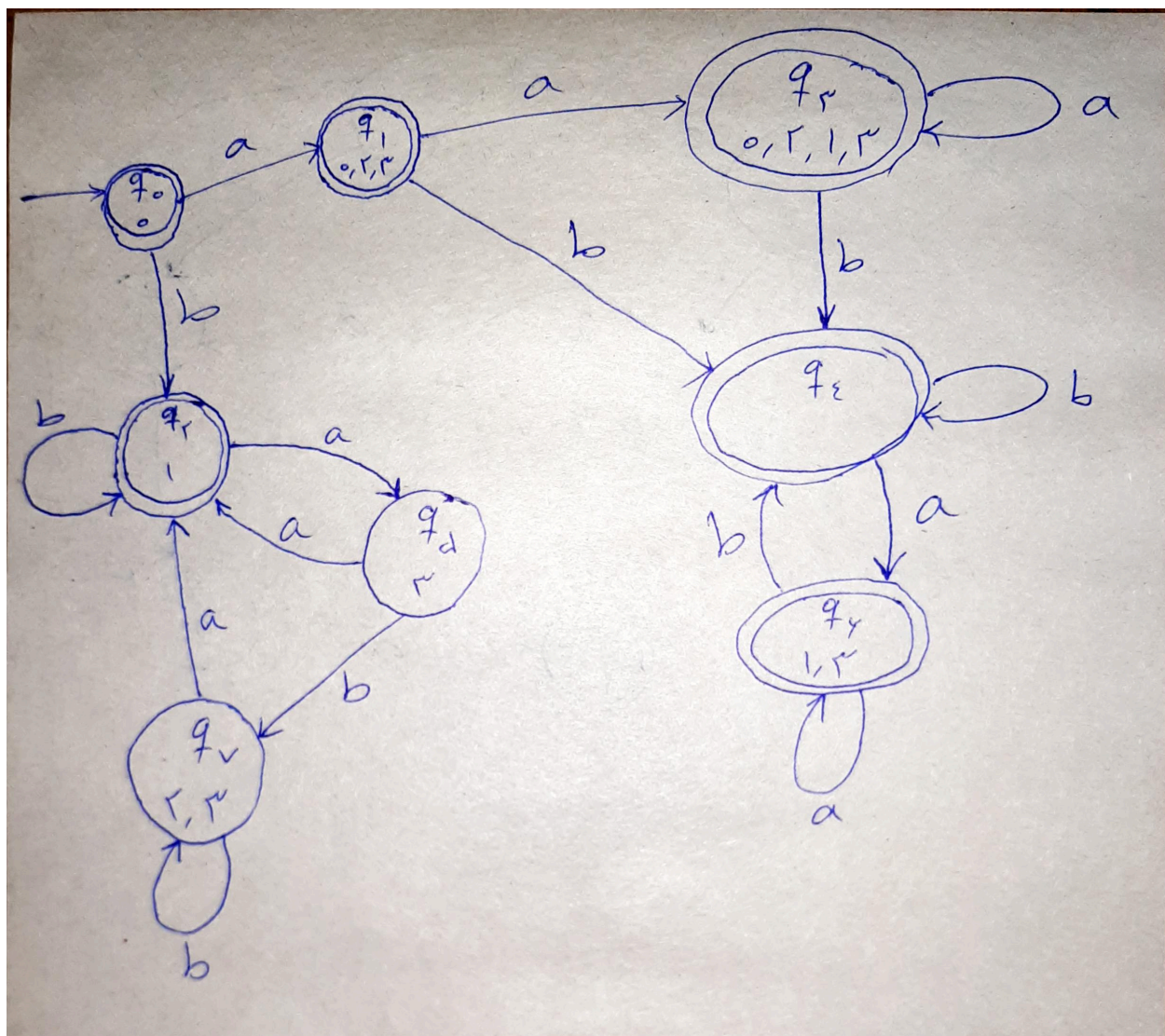


شکل ۳



شکل ۴

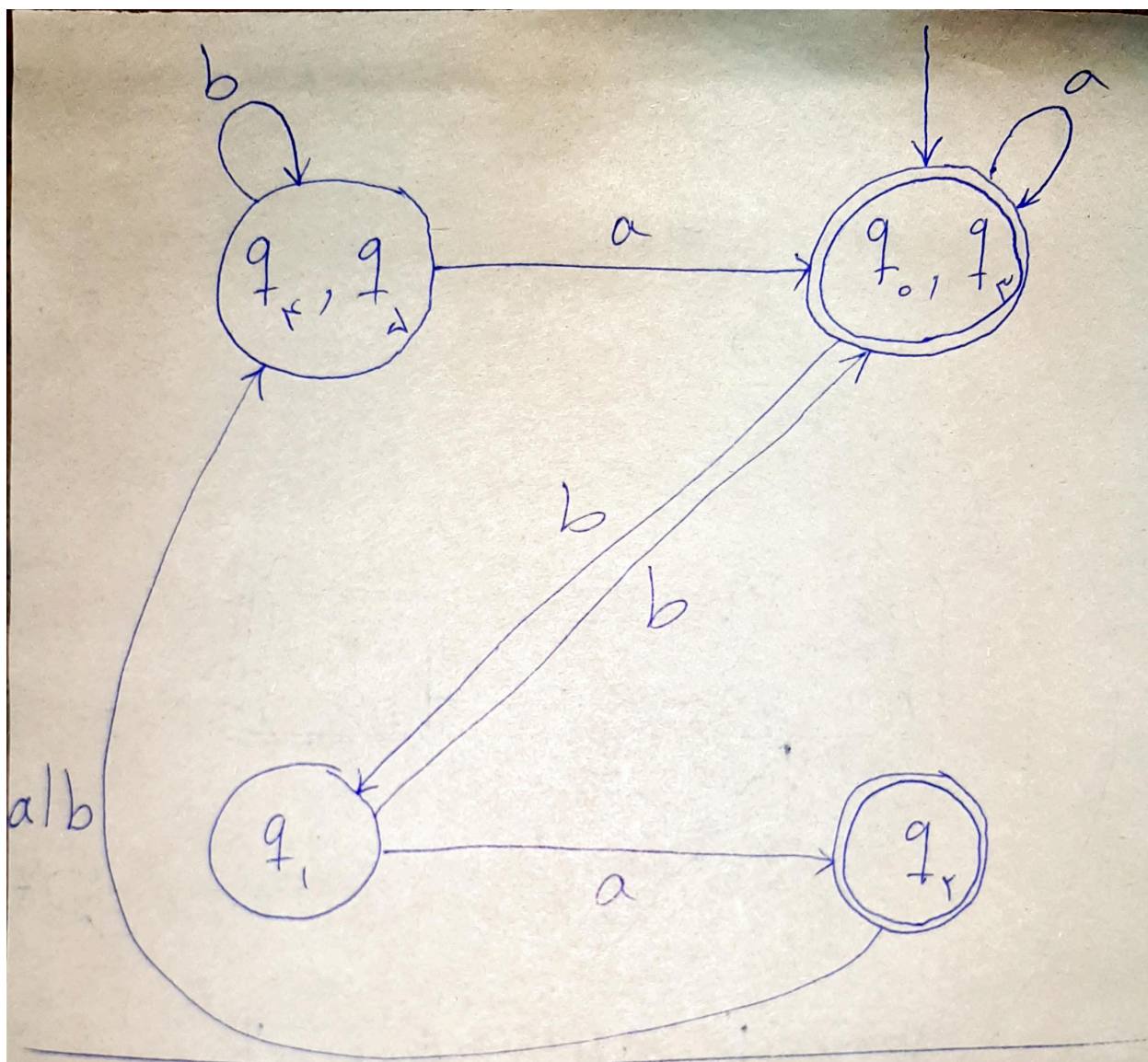
۲.۴ ج.



شکل ۵

۵

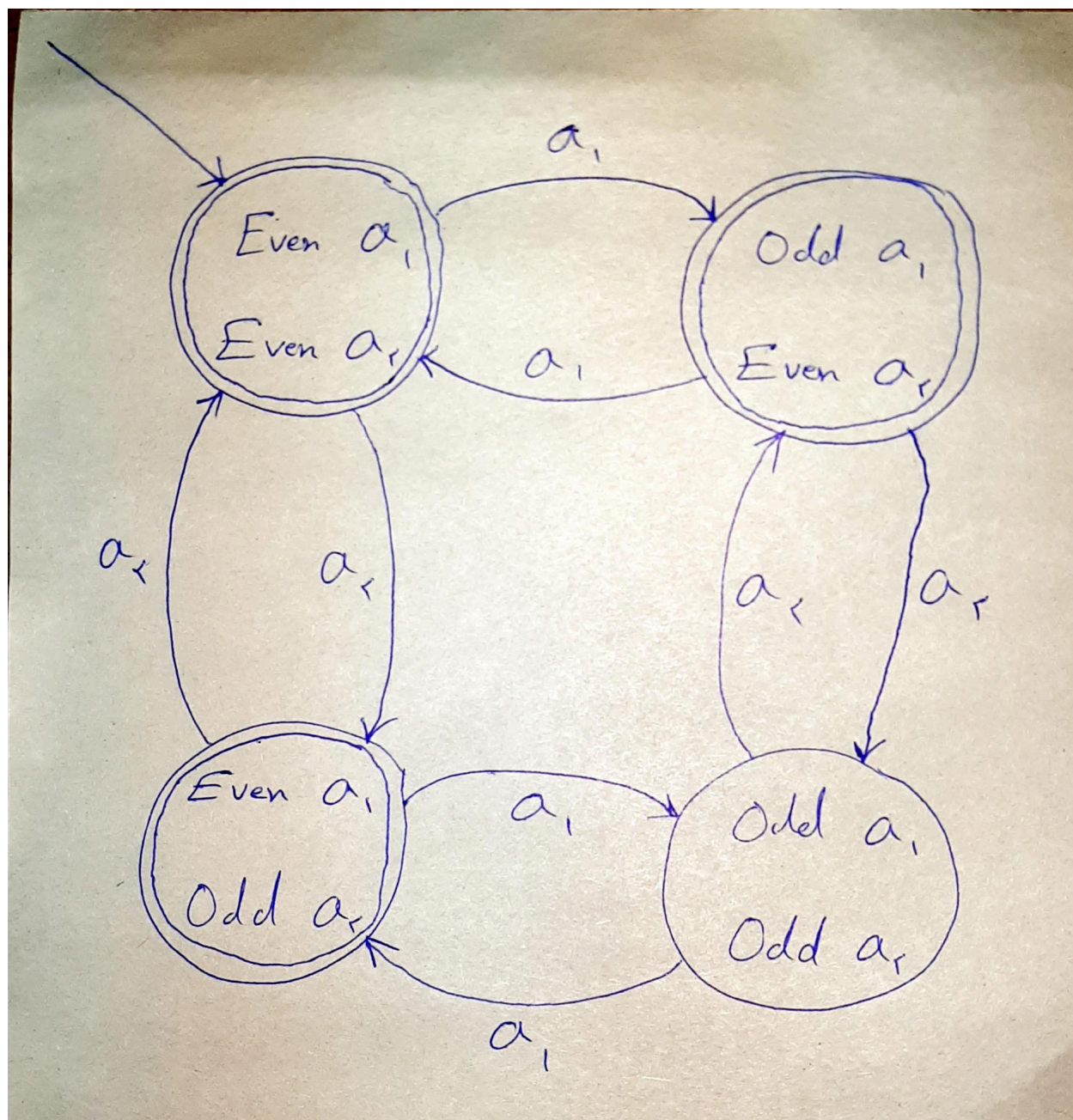
۱.۵ الف



شکل ۶

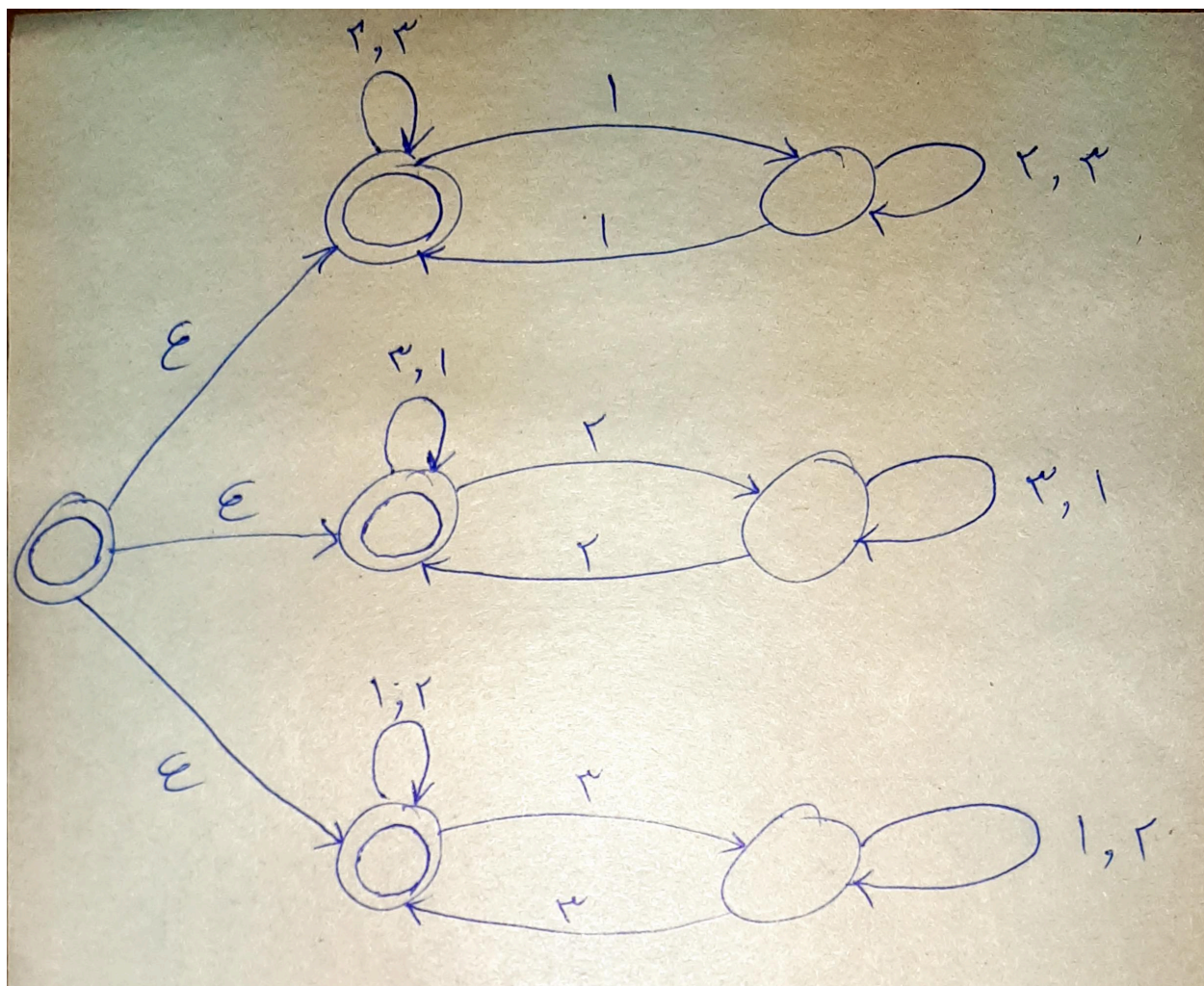
۲.۵ ب

قابل کمینه‌سازی نیست.



شکل ۷

۲.۶ ب



شکل ۸

۷

۱.۷ آ

برای زوج بودن عبارت $2n_a(w) + 3n_b(w)$ باید حتما زوج باشد. پس داریم:

$$(a \cup (bb))^*$$

۲.۷ ب

یا باید تعداد aها بزرگتر مساوی ۳ باشد یا تعداد bها بزرگتر مساوی ۲.

$$(aaa^+b^+) \cup (aa^+bb^+)$$

۳.۷ ج

از آنجایی که v تعداد حالات متناهی دارد هر یک از آنها را به دست آورده و در نهایت اجتماع می گیریم.

$$\begin{aligned}
 & a(a \cup b)^+ a \\
 & \cup b(a \cup b)^+ b \\
 & \cup aa(a \cup b)^+ aa \\
 & \cup ab(a \cup b)^+ ab \\
 & \cup ba(a \cup b)^+ ba \\
 & \cup bb(a \cup b)^+ bb \\
 & \cup aaa(a \cup b)^+ aaa \\
 & \cup aab(a \cup b)^+ aab \\
 & \cup aba(a \cup b)^+ aba \\
 & \cup abb(a \cup b)^+ abb \\
 & \cup baa(a \cup b)^+ baa \\
 & \cup bab(a \cup b)^+ bab \\
 & \cup bba(a \cup b)^+ bba \\
 & \cup bbb(a \cup b)^+ bbb \\
 & \cup aaaa(a \cup b)^+ aaaa \\
 & \cup aaab(a \cup b)^+ aaab \\
 & \cup aaba(a \cup b)^+ aaba \\
 & \cup aabb(a \cup b)^+ aabb \\
 & \cup abaa(a \cup b)^+ abaa \\
 & \cup abab(a \cup b)^+ abab \\
 & \cup abba(a \cup b)^+ abba \\
 & \cup abbb(a \cup b)^+ abbb \\
 & \cup baaa(a \cup b)^+ baaa \\
 & \cup baab(a \cup b)^+ baab \\
 & \cup baba(a \cup b)^+ baba \\
 & \cup babb(a \cup b)^+ babb \\
 & \cup bbaa(a \cup b)^+ bbaa \\
 & \cup bbab(a \cup b)^+ bbab \\
 & \cup bbba(a \cup b)^+ bbba \\
 & \cup bbbb(a \cup b)^+ bbbb
 \end{aligned}$$

۴.۷ د

برای فرد بودن طول رشته‌ها باید زوجیت تعداد a ها مخالف زوجیت تعداد b ها باشد.

$$(aa)^* a (bb)^* \cup (aa)^* (bb)^* b$$

۸

۱.۸

| # | Token | Type |
|----|----------|------------|
| 1 | main | ID |
| 2 | (| LPAREN |
| 3 |) | RPAREN |
| 4 | { | LBRACE |
| 5 | int | INT |
| 6 | * | STAR |
| 7 | a | ID |
| 8 | , | COMMA |
| 9 | b | ID |
| 10 | ; | SEMI |
| 11 | b | ID |
| 12 | = | ASSIGNMENT |
| 13 | 10 | NUM |
| 14 | ; | SEMI |
| 15 | a | ID |
| 16 | = | ASSIGNMENT |
| 17 | & | AMPERSAND |
| 18 | b | ID |
| 19 | ; | SEMI |
| 20 | printf | ID |
| 21 | (| LPAREN |
| 22 | ""%d%d"" | STRING |
| 23 | , | COMMA |
| 24 | b | ID |
| 25 | , | COMMA |
| 26 | * | STAR |
| 27 | a | ID |
| 28 |) | RPAREN |
| 29 | ; | SEMI |

| | | |
|----|---|------------|
| 30 | b | ID |
| 31 | = | ASSIGNMENT |
| 32 | * | STAR |
| 33 | b | ID |
| 34 | ; | SEMI |
| 35 | } | RBRACE |
| 36 | | EOF |

۲.۸

| # | Token | Type |
|----|--------|------------|
| 1 | main | ID |
| 2 | (| LPAREN |
| 3 |) | RPAREN |
| 4 | { | LBRACE |
| 5 | char | CHAR |
| 6 | ch | ID |
| 7 | = | ASSIGNMENT |
| 8 | 'A' | CHARACTER |
| 9 | ; | SEMI |
| 10 | int | INT |
| 11 | x | ID |
| 12 | , | COMMA |
| 13 | y | ID |
| 14 | ; | SEMI |
| 15 | x | ID |
| 16 | = | ASSIGNMENT |
| 17 | y | ID |
| 18 | = | ASSIGNMENT |
| 19 | 20 | NUM |
| 20 | ; | SEMI |
| 21 | x | ID |
| 22 | ++ | INCREMENT |
| 23 | ; | SEMI |
| 24 | printf | ID |

| | | |
|----|----------|--------|
| 25 | (| LPAREN |
| 26 | ""d% d"" | STRING |
| 27 | , | COMMA |
| 28 | x | ID |
| 29 | , | COMMA |
| 30 | y | ID |
| 31 |) | RPAREN |
| 32 | ; | SEMI |
| 33 | } | RBRACE |
| 34 | | EOF |

۳.۸

| # | Token | Type |
|----|---------|-----------------------|
| 1 | int | INT |
| 2 | strange | ID |
| 3 | (| LPAREN |
| 4 | int | INT |
| 5 | x | ID |
| 6 |) | RPAREN |
| 7 | { | LBRACE |
| 8 | if | IF |
| 9 | (| LPAREN |
| 10 | x | ID |
| 11 | <= | LESS THAN OR EQUAL TO |
| 12 | 0 | NUM |
| 13 |) | RPAREN |
| 14 | return | RETURN |
| 15 | 0 | NUM |
| 16 | ; | SEMI |
| 17 | if | IF |
| 18 | (| LPAREN |
| 19 | (| LPAREN |
| 20 | x | ID |
| 21 | % | MODULUS |

| | | |
|----|---------|-------------|
| 22 | 2 | NUM |
| 23 |) | RPAREN |
| 24 | != | NOTEQ |
| 25 | 0 | NUM |
| 26 |) | RPAREN |
| 27 | return | RETURN |
| 28 | x | ID |
| 29 | - | SUBTRACTION |
| 30 | 1 | NUM |
| 31 | ; | SEMI |
| 32 | return | RETURN |
| 33 | 1 | NUM |
| 34 | + | ADDITION |
| 35 | strange | ID |
| 36 | (| LPAREN |
| 37 | x | ID |
| 38 | - | SUBTRACTION |
| 39 | 1 | NUM |
| 40 |) | RPAREN |
| 41 | ; | SEMI |
| 42 | } | RBRACE |
| 43 | | EOF |

منابع

□□□□□□□□□□