

خطاهای lexical زمانی رخ میدهند که یک دنباله کاراکترها، با هیچ الگوی توکن تعریف شده ای در اسکنر همخوانی نداشته باشد، در این حالت lexical analyzer از ادامه اسکن باز میماند و باید از این حالت ریکاوری کند.

مثال های مختلف از این خطا

۱- مواجهه با کاراکتر غیرمجاز

```
#include <stdio.h>

int main() {
    puts("string");$
    return 0;
}
```

در این کد کاراکتر آخر خط چهارم غیر مجاز است

۲- عدم تطابق رشته با regex

```
#include <stdio.h>

int main() {
    /* unfinished comment
    puts("string");
    return 0;
}
```

در این کد، شروع کامنت چند خطی مشخص است اما پایان آن در کد موجود نیست

۳- جایگذاری یک کاراکتر، با یک کاراکتر غیر مجاز

```
#include <stdio.h>

int main() {
    int a = 12f3;
    return 0;
}
```

در این کد، در میان یک ثابت عددی، یک کاراکتر غیر عددی قرار دارد، در نتیجه این یک ارور lexical است.

```
#include <stdio.h>

int main() {
    float a = 123.g;
    return 0;
}
```

در اینجا به جای کاراکتر f که نشان دهنده یک ثابت عددی اعشاری است، کاراکتر g قرار گرفته که یک خطای نوشتاری است.

## 2

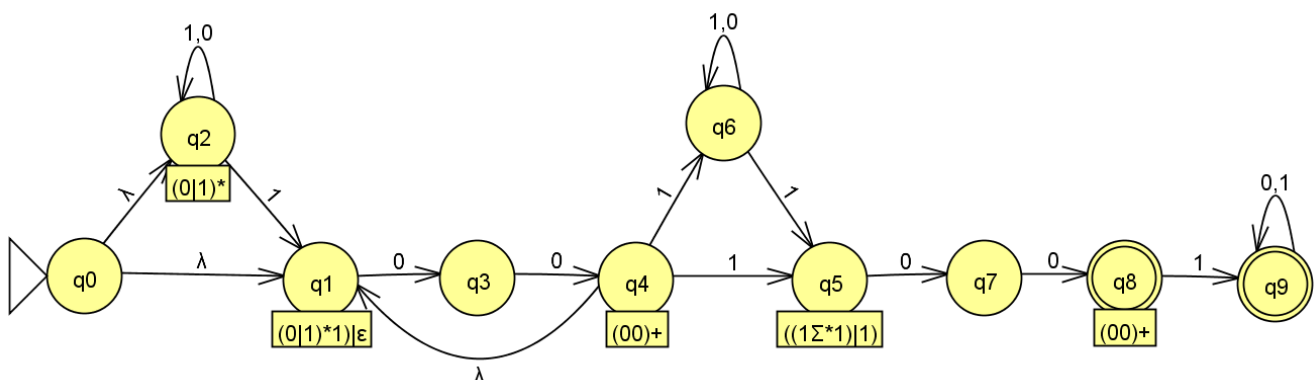
یک کامپایلر، یک برنامه را در زبان مبدا خوانده و به زبان مقصد ترجمه می‌کند و در صورت بروز خطا آن را گزارش می‌کند. در صورتی که زبان مقصد، زبان ماشین باشد، برنامه تولید شده می‌تواند اجرا و با ورودی‌های مورد نظر استفاده شود.

یک مفسر، به جای تولید یک برنامه در زبان مقصد به عنوان یک ترجمه، برنامه موجود در زبان سورس را مستقیماً بر روی ورودی‌های داده شده توسط کاربر اجرا می‌کند.

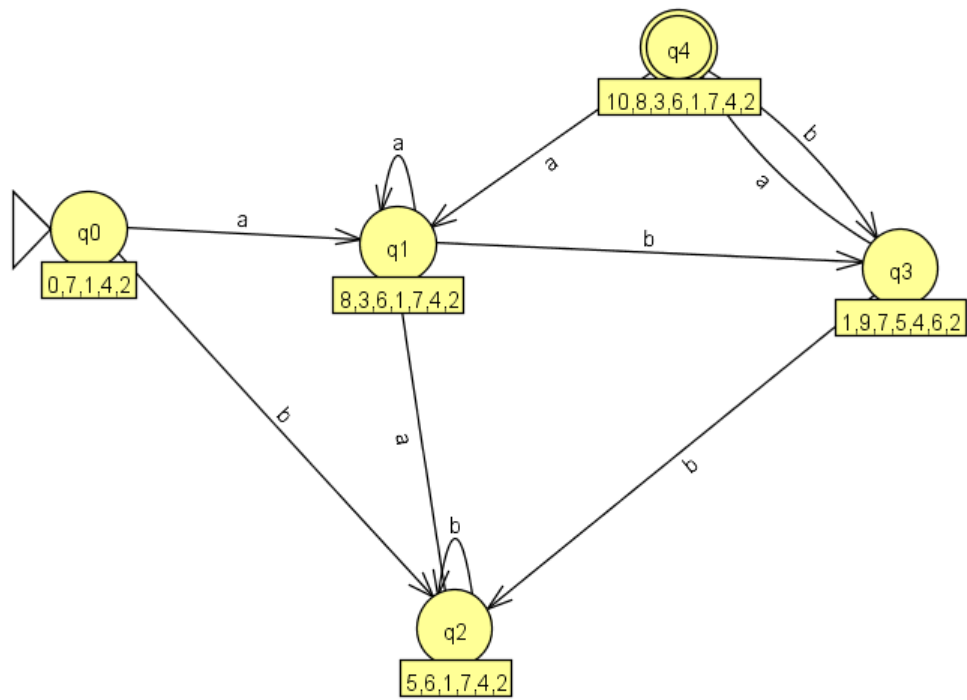
معمولاً برنامه‌های تولید شده توسط کامپایلر ها سریع تر از اجرای مفسر ها هستند، اما در مقابل، مفسر ها گزارش بهتری از خطاهای موجود در برنامه می‌دهند.

## 3

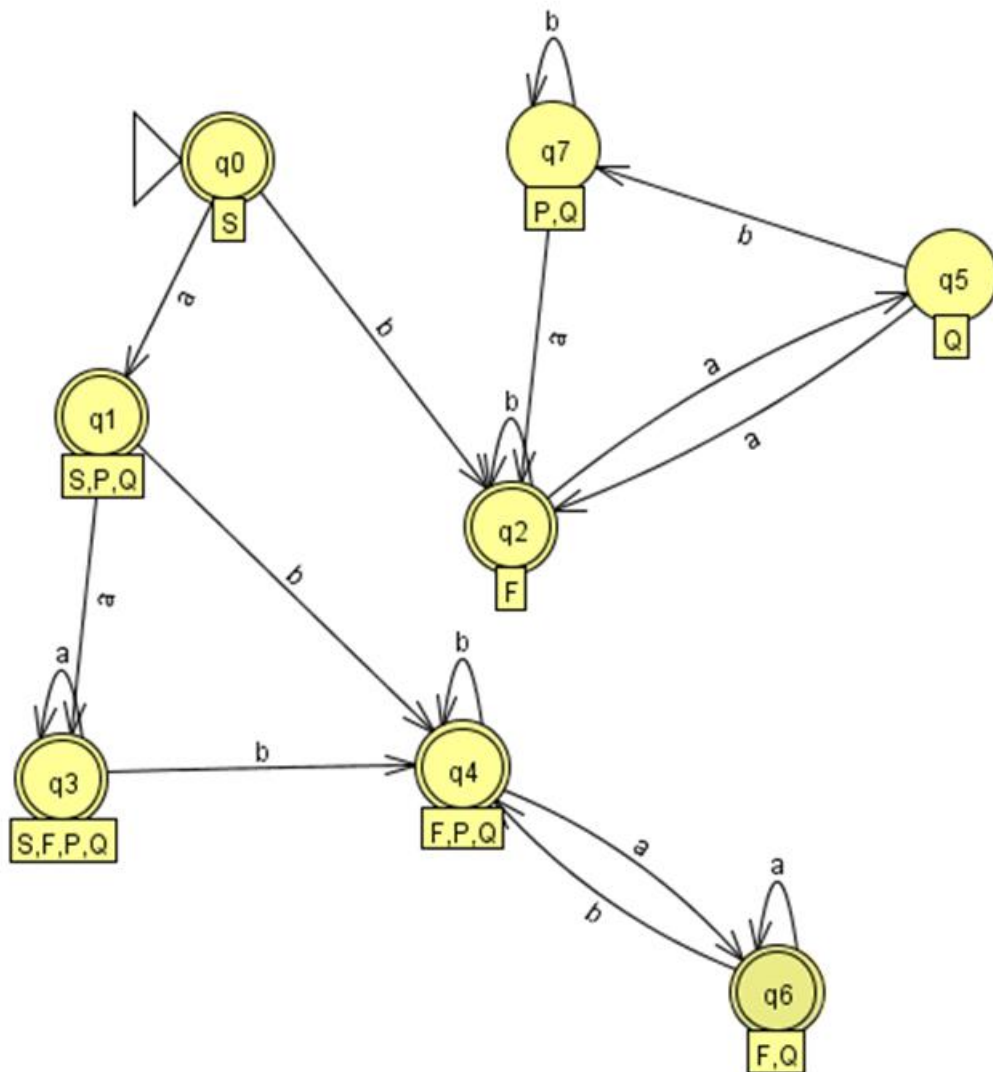
$$RE(L_1) = (\Sigma^*1)^*(00)^+((1\Sigma^*1)|1)(00)^+(1\Sigma^*)^*$$



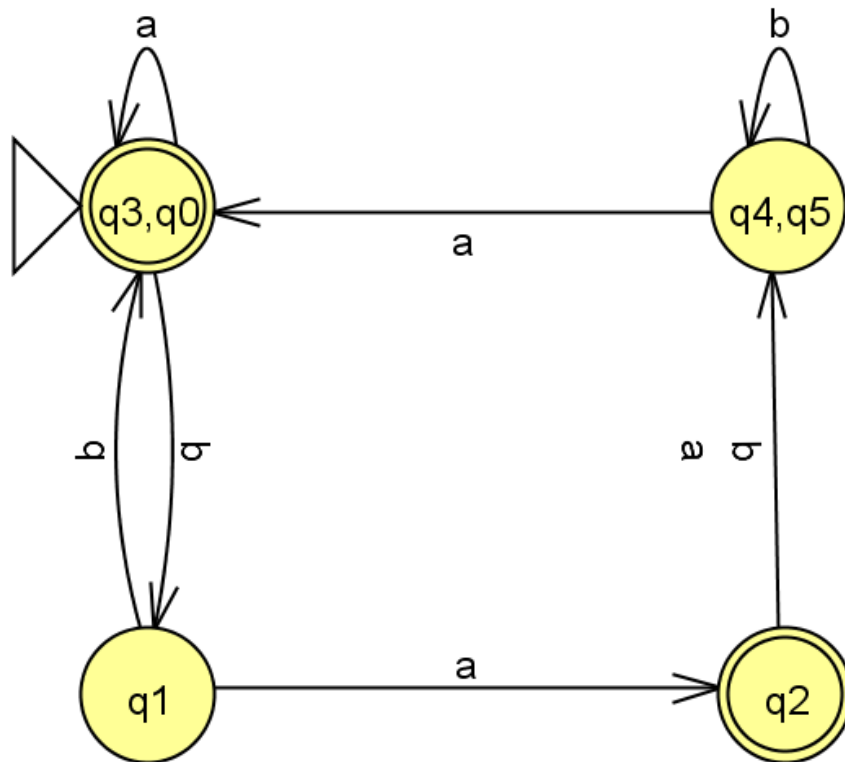
4-1



4-2



5-1



5-2

$$G_1 = \{1, 2, 7\}$$

$$G_2 = \{3, 4, 5, 6\}$$

State	a	b
1	2 G1	3 G2
2	4 G2	5 G2
7	7 G1	4 G2

State	a	b
3	6 G2	7 G1
4	5 G2	4 G2
5	7 G1	5 G2
6	2 G1	7 G1

$$G_1 = \{1, 7\}$$

$$G_2 = \{2\}$$

$$G_3 = \{3\}$$

$$G_4 = \{4\}$$

$$G_5 = \{5\}$$

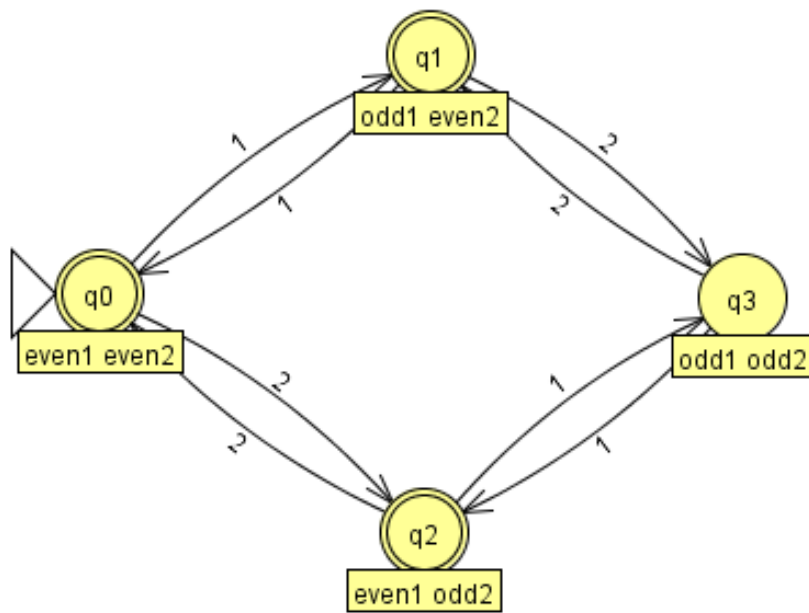
$$G_6 = \{6\}$$

State	a	b
1	2 G2	3 G3
7	7 G1	4 G4

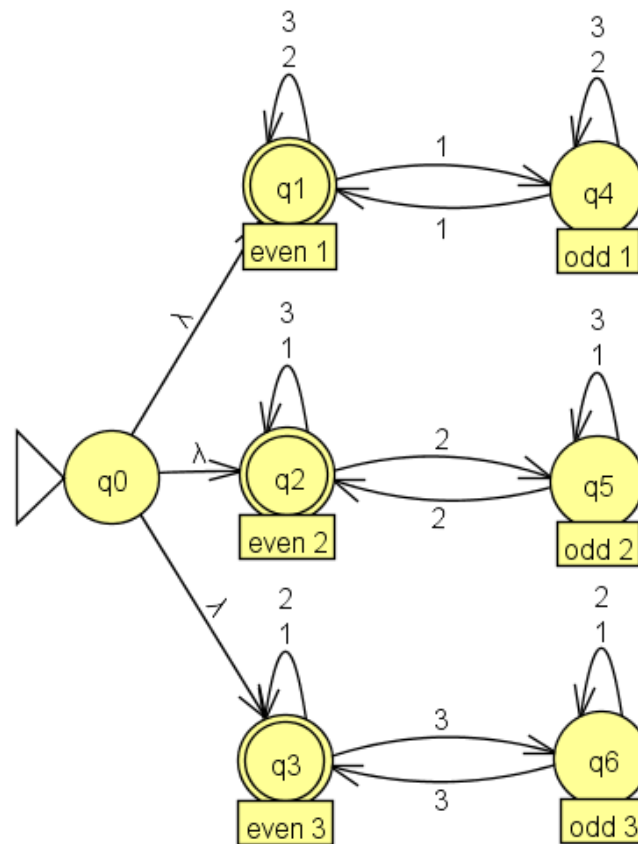
این DFA فشرده نمی شود.

با فرض اینکه تعداد تکرار صفر accept شود.

الف (  $\Sigma_2 = \{1, 2\}$  )



ب (  $\Sigma_3 = \{1, 2, 3\}$  )



الف) اگر تعداد  $b$  ها زوج باشد، شرط برقرار است.

$$RE(L_1) = (a | ba^*ba^*)^*$$

ب)

$$RE(L_2) = (aa^+bb^+) | (aaa^+b^+)$$

ج)

$$RE(L_3) = (a(a|b)^+a) | (b(a|b)^+b) | (aa(a|b)^+aa) | (ab(a|b)^+ab) | (ba(a|b)^+ba) | (bb(a|b)^+bb) | (aaa(a|b)^+aaa) | (aab(a|b)^+aab) | (aba(a|b)^+aba) | (abb(a|b)^+abb) | (baa(a|b)^+baa) | (bab(a|b)^+bab) | (bba(a|b)^+bba) | (bbb(a|b)^+bbb) | (aaaa(a|b)^+aaaa) | (aaab(a|b)^+aaab) | (aaba(a|b)^+aaba) | (aabb(a|b)^+aabb) | (abaa(a|b)^+abaa) | (abab(a|b)^+abab) | (abba(a|b)^+abba) | (abbb(a|b)^+abbb) | (baaa(a|b)^+baaa) | (baab(a|b)^+baab) | (baba(a|b)^+baba) | (babb(a|b)^+babb) | (bbaa(a|b)^+bbaa) | (bbab(a|b)^+bbab) | (bbba(a|b)^+bbba) | (bbbb(a|b)^+bbbb)$$

د)

$$RE(L_4) = (aa)^*(a|b)(bb)^*$$

الف)

ID(main)	LPAREN	RPAREN	LBRACE	INT	STAR	ID(a)	COMMA	ID(b)	SEMI
ID(b)	ASSIGN	NUM(10)	SEMI	ID(a)	ASSIGN	AND	ID(b)	SEMI	ID(sprintf)
LPAREN	STRING(%d%d)	COMMA	ID(b)	COMMA	STAR	ID(a)	RPAREN	SEMI	ID(b)
ASSIGN	ID(b)	SEMI	RBRACE	EOF					

ب) فرض: در این اسکندر، کاراکترها به صورت عدد اسکی توکن می شوند

ID(main)	LPAREN	RPAREN	LBRACE	CHAR	ID(ch)	ASSIGN	NUM(41)	SEMI	INT
ID(x)	COMMA	ID(y)	SEMI	ID(x)	ASSIGN	ID(y)	ASSIGN	NUM(20)	SEMI
ID(x)	INC	SEMI	ID(sprintf)	LPAREN	STRING(%d %d)	COMMA	ID(x)	COMMA	ID(y)
RPAREN	SEMI	RBRACE	EOF						

ج)

INT	ID(strange)	LPAREN	INT	ID(x)	RPAREN	LBRACE	IF	LPAREN
ID(x)	LESSEQ	NUM(0)	RETURN	NUM(0)	SEMI	IF	LPAREN	LPAREN
ID(x)	PERCENT	NUM(2)	RPAREN	NOTEQ	NUM(0)	RPAREN	RETURN	ID(x)
MINUS	NUM(1)	RETURN	NUM(1)	PLUS	ID(strange)	LPAREN	ID(x)	MINUS
NUM(1)	RPAREN	SEMI	RBRACE					