جلسه سوم

دیاگرام تشخیص عملگر های رابطه ای

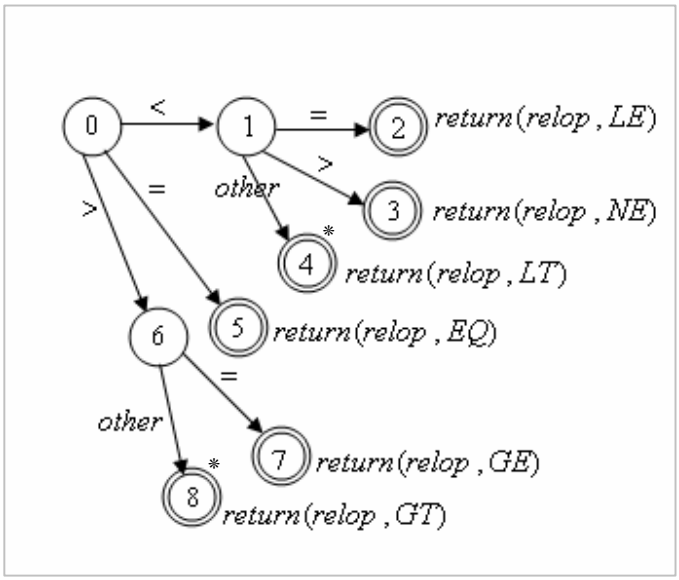
عملگر های رابطه ای =< , < , <> , = , >= , >

$L : lese$

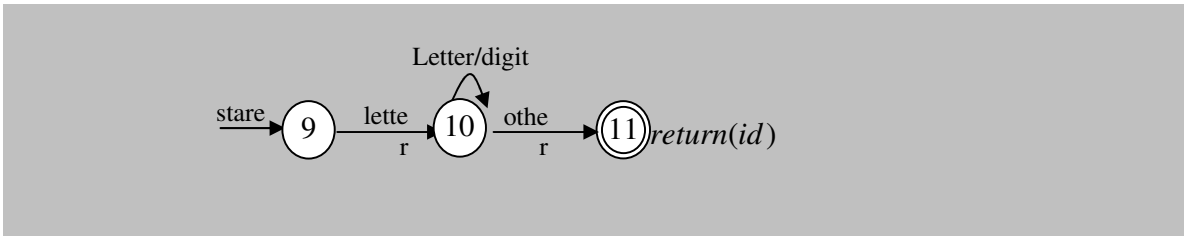
$E : Equal$

$G : grather$

کاراکتر خوانده شده بایستی به بافر برگردانده شود.



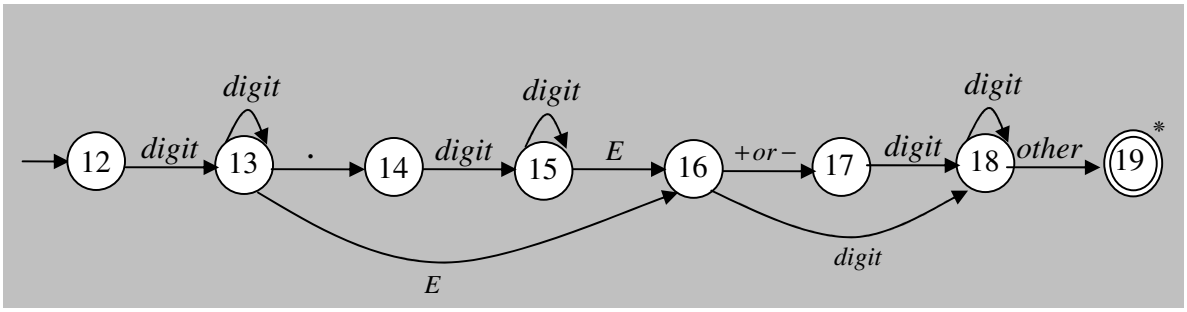
دیاگرام تشخیص شناسه(identifire).



دیاگرام تشخیص اعداد

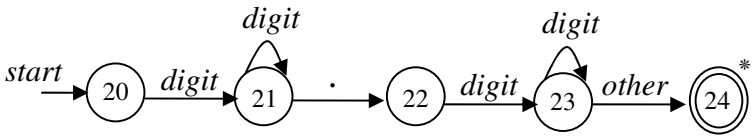
تشخیص اعداد به صورت نماد علمی

$digit\,.\,digit\ \ E\ \ (+/-)\ \ digit$	$12.34E+1$
$digit\,.\,digit\ \ E\ \ digit$	$12.34\,E1$
$digit\ \ E\ \ (+/-)\ \ digit$	$12E+1$
$digit\ \ E\ \ digit$	$12E1$

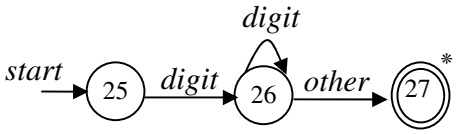


تشخیص اعداد به صورت اعشاری

digit.digi



تشخیص اعداد به صورت صحیح



برنامه به زبان c برای پیدانمودن حالت شروع بعدی

```
int state=0 , start=0;
int  fail()
{
switch (state) {
case 0 :     start = 9:break;
case 9 :     start =12:break;
case 12 :    start = 20:break;
case 20 :    start = 25:break;
case 25 :    recover():break;
default :    /* compiler  error /*
}
return  start;
}
```

Default

رشته خوانده شده با الگوی هیچ کدام از توکن ها مطابقت نمی کند در این صورت خطا گزارش می شود ولی کامپایلر نباید متوقف شود بلکه اسکن بقیه رشته ها را ادامه می دهد.

برنامه به زبان c برای تحلیگر لغوی در صفحه بعد

```
while (1){
  switch (state){
case 0: c = nextchar(); if (isspace(c)){lexeme _ beginnin++ , state = 0;}
      else if (c =='<')   state = 1;
      else if (c =='=')   state = 5;
      else if (c =='>')   state = 6;
      else state = fail();
      break;
case 1  c = nextchar();
      if (c =='=')   state = 2;
     else if (c =='>')   state = 3;
     else state = 4;
    break;
case 2 :  tokentype = relop; tokenvalue = LE; break;
case 3 :  tokentype = relop; tokenvalue = NE; break;
case 4 :  ungetch(c); tokentype = relop; tokenvalue = L; break;
⋮
case 9 : c = nextchar();
       if (isletter(c))    state = 10;
       else   state = fail();
      break;
case 10: c = nextchar();
       if (isletter(c) || isdigit(c))   state = 10;
      else   state = 11;   break;
case 11: ungetch(c); install(token value);
      return(gettoken(token value));
     break;
case 12: c = nextchar();
        if (isdigit(c))   state = 13;
        else   state = fail();
       break;
case 13: c = nextchar();
        if (isdigit(c))   state = 13;
        else if (c ='.')   state = 14;
        else if (c ='E')   state = 16;
        else   state = fail();
       break;
case 14: c = nextchar();
        if (isdigit(c))   state = 15
        else   state = fail();
       break;
case 15: c = nextchar();
        if (isdigit(c))   state = 15
        else if (c ='E')   state = 16;
        else   state = fail();
       break;
```

توضیحات: تابع install() به جدول سمبل ها نـگاه می کند اگر Token خوانده شده کلمه کلیدی باشد صفر را بر می گرداند اگر کلمه کلیدی نباشد و شناسه باشد دو حالت داریم اگر این شناسه در جدول سمبلها موجود باشد اشاره گر به مدخلی که ایـن توکن در آن قرار دارد برگردانده میشود و اگر در جدول توکن ها موجود نباشد در جدول سمبل ها قرار داده میشود و اشاره گر به مدخلی که قرار داده میشود بر گردانده می شود(شکل زیر جدول سمبل).

جدول سمبل ها

If	k
wa	id
a2	id

☐ اگر برای تشخیص کلمات کلیدی از دیاگـرام انتقال حالت استفاده شود، تعداد حالات زیاد شده و از این رو state گردانی مشکل خواهد بود و سرعت پایین میاید ، پس بهتر است کلمات کلیدی در آغاز کار در جدول سمبلها یا جدول نشانه ها قرار داده شوند

☐ نام تابع شناسه است

☐ تابع gettoken() اگر توکن خوانده شده کلمه کلیدی باشد نشانه متناظر آن برگردانده میشود و اگر شناسه باشد id را برمیگرداند.

☐ 9 , 10 , 11 برای تشخیص کلمات کلیدی است.

☐ بهتر است که توکن هائی که بیشتر تکرار می شوند، در state های آغازین قرار گیرند، یعنی در case های اول که باعث می شود سرعت بالا رود.

ادامه برنامه قبلی

```
cace 16: c = nextchar();
        if (isdigit(c))    state = 18;
        else if (c ='+'|| c ='–')   state = 17;
        else   state = fail();
        break;
cace 17: c = nextchar();
        if (isdigit(c))    state = 18;
        else   state = fail();
        break;
cace 18: c = nextchar();
        if (isdigit(c))    state = 18;
        state = 19
        break;
cace 19: ungetch(c); tokentype ='num'; tokenvalue =     ;
        if (isdigit(c))    state = 18;
        break;
cace 20: c = nextchar();
        if (isdigit(c))    state = 21;
        else   state = fail();
        break;
cace 21: c = nextchar();
        if (isdigit(c))    state = 21;
        else if (c ='.')   state = 22;
        else   state = fail();
        break;
cace 22: c = nextchar();
        if (isdigit(c))    state = 23;
        else   state = fail();
        break;
cace 23: c = nextchar();
        if (isdigit(c))    state = 23;
        else   state = 24;
        break;
cace 24: c = ungetch(c); tokentype ='num'; tokenvalue =     ;
cace 25: c = nextchar();
        if (isdigit(c))    state = 26;
        else   state = fail();
        break;
cace 26: c = nextchar();
        if (isdigit(c))    state = 26;
        else   state = 27;
        break;
cace 27: ungetch(c); tokentype ='num'; tokenvalue =     ;
}
}
```