



# الگوریتم های تست اتوماتیک نرم افزار

نام و نام خانوادگی: مینا بیرامی  
استاد راهنما: دکتر الهام محمودزاده  
مهندسی کامپیوتر\_فناوری اطلاعات  
مهرماه ۱۳۹۷

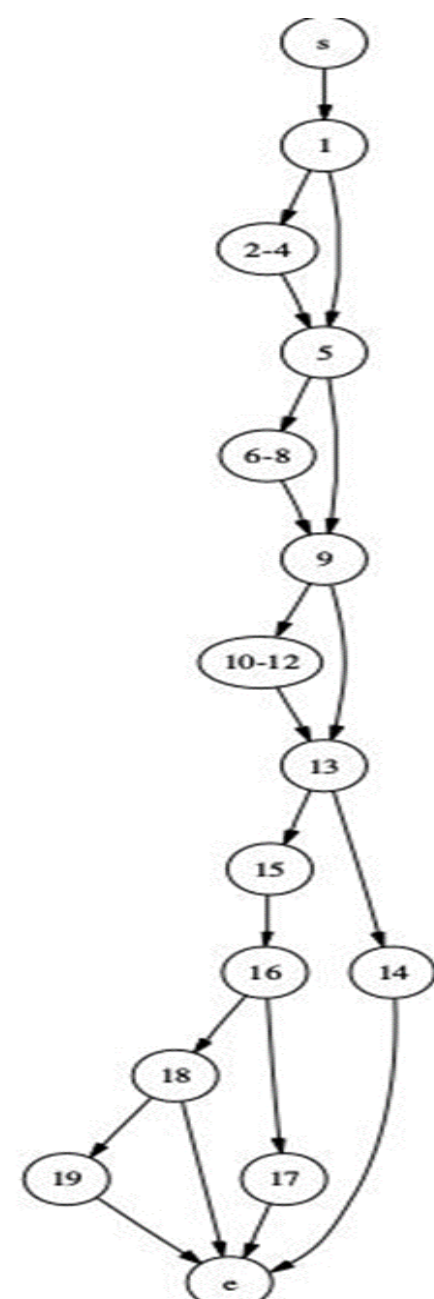
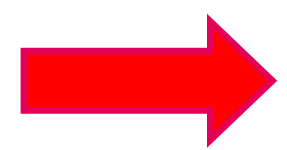
## ۱.مقدمه

تست اتوماتیک نرم افزار و روش های سریع رسیدن به جواب در این تست ها ، همواره مورد توجه محققان در طول سال های متمادی بوده است  
در این پژوهش ، قصد ما بررسی برخی از الگوریتم های حل مساله مورد نظرمان از جمله الگوریتم ژنتیک ، الگوریتم ژنتیک ، ماشین بردار پشتیبان و درخت تصمیم گیری است.  
در حوزه white-box یا همان تست ساختاری ، با بررسی ما ساختار برنامه ، قواعدی که از آن استناد میکنیم و به صورت اتوماتیک به تولید مسیر های که به تولید داده های تست منجر می شوند را تولید کنیم.

## ۲.بررسی مساله و گراف کنترل جریان

گراف کنترل جریان ، گرافی است که هر خط برنامه را یک گره در نظر میگیرد و هر گره را به گره هایی که در اجرای برنامه ملاقات می کند ،متصل می کند. برنامه به صورت زیر است:

```
s      int tri_type(int a, int b, int c)
      {
          int type;
          if (a > b)
          {   int t = a; a = b; b = t;   }
          if (a > c)
          {   int t = a; a = c; c = t;   }
          if (b > c)
          {   int t = b; b = c; c = t;   }
          if (a + b <= c)
          {
              type = NOT_A_TRIANGLE;
          }
          else
          {
              type = SCALENE;
              if (a == b && b == c)
              {
                  type = EQUILATERAL;
              }
              else if (a == b || b == c)
              {
                  type = ISOSCELES;
              }
          }
          return type;
      }
```



در ادامه به تناقضات موجود در برنامه می پردازیم .شرط های خطوط ۱ و ۵ و ۹ را در جدول زیر با روابط ریاضی نوشتیم.با نوشتن روابط ریاضی موجود در ستون آخر کنار هم از ۸ حالت موجود ، ۳ حالت به تناقض خواهد رسید .بنابراین تعداد مسیر های برنامه کاهش خواهد یافت

با نوشتن روابط ریاضی آنها

وضعیت های S1S4S6 و

S1S3S6 و S2S3S6 هیچ گاه

رخ نخواهد داد .بنابراین تعداد

مسیر ها از ۳۲ به ۲۰ کاهش می

یابد.

## ۳.تولید مسیر های باقی مانده با استفاده از الگوریتم ها

قصد تولید مسیر های صحیح با قی مانده را داریم. ۲ متود مطرح را برای این امر برگزیده ایم:

### •الگوریتم تکاملی :لگوریتم ژنتیک:

- تعریف ساختار مسیر مسیر ها با طول های متفاوت: شماره نود های پیموده شده در مسیر ذخیره میشود
- جمعیت اولیه : تولید به صورت تصادفی
- شرط خاتمه : رسیدن به ۲۰ مسیر صحیح
- تابع fitness function: به ازای هر اتصال صحیح گره به گره بعدیش داریم:
  - Fitness value+1 = string's length

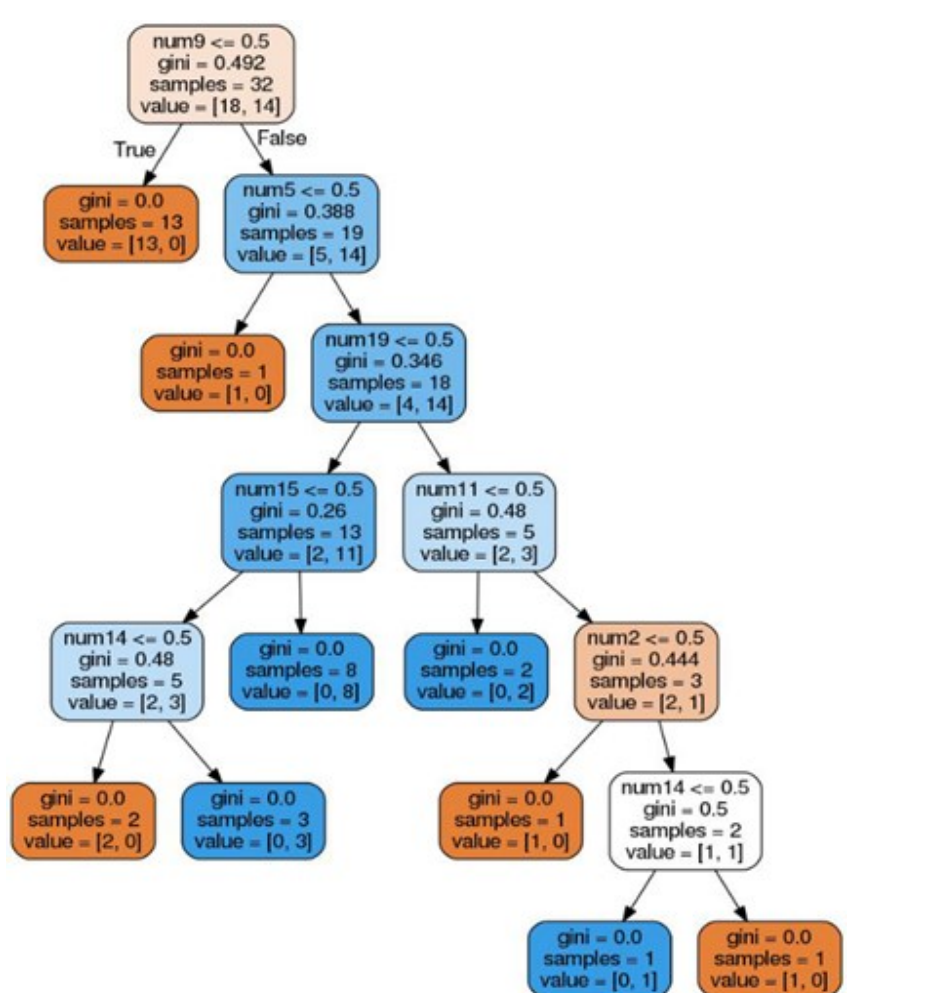
- تابع CROSS-OVER: مسیر های با امتیاز بیشتر از نقطه ۱۳ که مرکز ثقل برنامه است ، با هم جابجا می شوند.
- تابع mutation: جهش های خودکار با استفاده از مطالعه ساختار برنامه در نود های اساسی از جمله ۱ و ۵ و ۹ و ...

### • الگوریتم های یادگیری : SVM و Decision Tree

- تعریف ساختار مسیر : ساختار دودویی : تمامی مسیر ها دارای ۲۰ زن ۰ و ۱ هستند . ۱ به معنای حضور در مسیر و ۰ به معنای عدم حضور آن است.
- ماتریس سردرگمی و تعریف معیار های سنجش آن از جمله Precision و recall.....
- تهیه دیتا ست در یک فایل: دارای ۵۰ رکورد شامل ۲۰ مسیر صحیح و ۳۰ مسیر اشتباه
- تقسیم داده های تمرین و تست : به نسبت ۳۰-۷۰
- اجرای الگوریتم SVM با هسته گوسی (RBF Kernel)
- اجرای الگوریتم decision tree و استخراج نود های مهم

## ۴.مشاهده نتایج

- الگوریتم ژنتیک: پس از تکرار مراحل تولید مسیر های صحیح به مدت ۵۰ بار
- الگوریتم SVM: این الگوریتم بر روی دیتا ست با random state های مختلف در ۲۰ ایتريشن امتحان شد
- الگوریتم Decision Treee: این الگوریتم بر روی دیتا ست با random state های مختلف در ۲۰ ایتريشن امتحان شد.



	precision	recall	f1-score	support
0	0.89	0.89	0.89	9
1	0.83	0.83	0.83	6
avg / total	0.87	0.87	0.87	15

	precision	recall	f1-score	support
0	1.00	1.00	1.00	9
1	1.00	1.00	1.00	6
avg / total	1.00	1.00	1.00	15

	precision	recall	f1-score	support
0	0.88	1.00	0.93	7
1	1.00	0.88	0.93	8
avg / total	0.94	0.93	0.93	15

1: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 20]  
2: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 16, 17, 20]  
3: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 16, 18, 19, 20]  
4: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 16, 18, 20]  
5: [0, 1, 2, 3, 4, 5, 9, 13, 14, 20]  
6: [0, 1, 2, 3, 4, 5, 9, 13, 15, 16, 17, 20]  
7: [0, 1, 2, 3, 4, 5, 9, 13, 15, 16, 18, 19, 20]  
8: [0, 1, 2, 3, 4, 5, 9, 13, 15, 16, 18, 20]  
9: [0, 1, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 20]  
10: [0, 1, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 16, 17, 20]  
11: [0, 1, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 16, 18, 19, 20]  
12: [0, 1, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 16, 18, 20]  
13: [0, 1, 5, 9, 10, 11, 12, 13, 14, 20]  
14: [0, 1, 5, 9, 10, 11, 12, 13, 15, 16, 17, 20]  
15: [0, 1, 5, 9, 10, 11, 12, 13, 15, 16, 18, 19, 20]  
16: [0, 1, 5, 9, 10, 11, 12, 13, 15, 16, 18, 20]  
17: [0, 1, 5, 9, 13, 14, 20]  
18: [0, 1, 5, 9, 13, 16, 17, 20]  
19: [0, 1, 5, 9, 13, 15, 16, 18, 19, 20]  
20: [0, 1, 5, 9, 13, 15, 16, 18, 20]

چند نمونه از گزارش های به کار گیری

الگوریتم ژنتیک : تولید ۲۰ مسیر صحیح که پس از ۵۰ بار تکرار به طور میانگین در طول ۱۸ بار تولید جمعیت جدید به مجموعه جواب می رسید.  
الگوریتم SVM: در هر کدام از انها مقدار precision و recall و به تبع آن دو f1-score تغییر میکند ولی تغییرات آن چنان مشهود نیست و در راج نسبتا بالایی (نزدیک به ۹۰ درصد) قرار دارد . بنابراین میتوان اذعان داشت الگوریتم تا حد بسیار خوبی درست و به جا استفاده شده است.

یک نمونه درخت ایجاد شده : کاهش ۲۰ ویژگی

به ۶ ویژگی مهم

ریشه درخت گره شماره ۹

سایر ویژگی های الویت دار به ترتیب عبارت

است از :

گره شماره ۵، ۱۹، ۱۵، ۱۱، ۲، ۱۴

## ۵.نتیجه گیری

الگوریتم ژنتیک با وجود سابقه طولانی اش ، همچنان یکی از مهم ترین الگوریتم های حال حاضر شناخته می شود . تولید اتوماتیک مسیر های برنامه ما با الگوریتم ژنتیک با تعداد محدودی تکرار در مدت زمان قابل قبولی ، با موفقیت رو به رو شد . **Fitness function** ارائه شده در عین سادگی ، معیار قابل قبولی برای تست این برنامه به شمار آمد.  
الگوریتم های **Decision Tree** و **SVM** از الگوریتم های معروف طبقه بندی ، نتایج با دقت بالایی به ارمغان آوردند . نکته قابل ملاحظه ، زمان بسیار کوتاه در کشف الگو های دیتاست است. الگو ها و قوانین بدست آمده در نتایج مطابقت نزدیک به ۱۰۰ درصد به الگوهای واقعی داشتند. اما کمبود داده ها به علت کوتاه بودن قطعه کد ، ممکن است در نتایج بسیار خوب این الگوریتم ها بی تاثیر نباشد .