



کانال مہمات شریف

  @SHARIF_IE



طراحی کامپایلرها

نیم‌سال دوم ۹۸-۹۹

مدرس: محمدرضا بهرامی

زمان امتحان: ۱۶۵ دقیقه

۱۴ تیر ۹۹، ۶۰ + ۶۰ نمره

آزمون پایان‌ترم

درستی یا نادرستی عبارات زیر را مشخص کنید. هر پاسخ غلط به اندازه نمره همان سوال، امتیاز منفی خواهد داشت.

- (۱ نمره) ۱. در یک گراف نحو^۱، از یک رأس می‌توان با حداکثر یک یال که نام آن، نام یک گراف باشد بیرون آمد.
- (۱ نمره) ۲. ارسال آرگومان اضافی به یک تابع نوعی خطای نحوی^۲ است.
- (۱ نمره) ۳. برای ترجمه‌ی یک گرامر L-خصیصه‌دار^۳ محاسبه‌ی مقادیر ویژگی‌های موروثی واژه‌ی نحوی سمت چپ، در آخر آن قاعده انجام می‌شود.
- (۱ نمره) ۴. تعداد آرگومان‌های یک تابع و نوع آن‌ها در جدول نمادها نگهداری می‌شود.
- (۱ نمره) ۵. پیوند دسترسی^۴، اشاره‌گری به تابع فراخواننده است.
- (۱ نمره) ۶. در میان روش‌های ارسال پارامتر به تابع، فراخوانی با ارجاع^۵ عملاً با فراخوانی مقدار-نتیجه^۶ یکسان است.
- (۱ نمره) ۷. مرتب کردن دستورات بر اساس ترتیب توپولوژیک و پیمایش عمق اول آن، زمان‌بندی بهینه‌ی دستورات را برای اجرا بر روی یک ماشین مجهز به خط لوله^۷ تولید می‌کند.
- به سوالات زیر پاسخ کوتاه بدهید.
- (۱ نمره) ۸. هنگام عبور از یالی که برجسب یک گراف دارد و یک نام یک روال مفهومی^۸ نیز بر روی آن قرار دارد، این روال باید قبل از فراخوانی گراف اجرا شود یا پس از آن؟ چرا؟
- (۱ نمره) ۹. چه زمانی فرآیند ساختاریابی^۹ در گراف نحو با موفقیت به پایان می‌رسد؟
- (۲ نمره) ۱۰. آیا همواره مناسب است تا دستورات مستقل از حلقه را از حلقه خارج کنیم؟ این مورد را از نظر صحت و بهینه‌سازی بررسی کنید.
- (۱ نمره) ۱۱. چه زمانی اضافه کردن ناپایانه‌های نشان‌دار^{۱۰} در پارسرهای LR، برای ترجمه‌ی هدایت‌شده با نحو^{۱۱}، باعث ایجاد تداخل می‌شود؟ چه تداخلی؟
- (۱ نمره) ۱۲. یک ویژگی نام ببرید که فقدان آن منجر به دشوار شدن پیاده‌سازی کامپایلر تک‌گذره^{۱۲} می‌شود.
- (۱ نمره) ۱۳. چرا کامپایلرها از چند سطح کد میانی استفاده می‌کنند؟
- (۲ نمره) ۱۴. به طور خلاصه، کلیت ایده‌ی خود جهت چگونگی پیاده‌سازی یک Coroutine را بیان کنید. توجه داشته باشید ممکن است چند Coroutine هم‌زمان در حال اجرا باشند.

- Syntax Graph^۱
Syntax Error^۲
L-Attributed^۳
Access Link^۴
Call-By-Reference^۵
Call-By-Value/Result^۶
Pipeline^۷
Semantic Routine^۸
Parsing^۹
marked nonterminals^{۱۰}
Syntax Directed Translation^{۱۱}
Sigle Pass^{۱۲}

۱۵. برای تولید کد دستور break داخل حلقه، چه باید کرد؟ (با توجه به کد تولیدی در اسلاید تولید کد سه آدرسه، به طور خلاصه توضیح دهید). (۲نمره)

۱۶. بهینه‌سازی به روش انتشار کپی^{۱۳} چگونه باعث بهینه شدن کد میانی می‌شود؟ (۱نمره)

۱۷. در برخی از زبان‌های برنامه‌سازی مانند C/C++، هنگامی که یک متغیر مقداردهی اولیه نمی‌شود، به برنامه‌نویس اخطار داده می‌شود. این اخطار را از منظر بهینه‌سازی بررسی کنید. (۱نمره)

در اسلایدهای سری ۴ بخش ۲ صفحه‌ی ۱۵۰، یک پارسر LALR ارائه شده است^{۱۴}. همچنین فرض کنید که طی یک تحقیق متوجه شدیم که برنامه‌نویسان معمولاً تساوی را فراموش می‌کنند. به سوالات ۱۸، ۱۹ و ۲۰ پاسخ دهید:

۱۸. چه روشی را برای خطایابی پیشنهاد می‌کنید؟ چرا؟ (۱نمره)

۱۹. با انجام چه تغییراتی در جدول ساختاریابی^{۱۵} این روش پیاده‌سازی می‌شود؟ (۲نمره)

۲۰. اگر مشابه همین مشکل با کاراکتر ستاره وجود داشت، وضعیت چگونه بود؟ (۱نمره)

۲۱. گرامر زیر برای تولید اعداد اعشاری در مبنای ۸ نوشته شده است. یک SDD که L-خصیصه‌دار باشد برای محاسبه‌ی مقدار عددی اعشار ارائه دهید. برای ارائه این تعریف نباید از حلقه استفاده کنید ولی شرط مجاز است. (۵نمره)

$$\begin{aligned} N &\rightarrow L . L \\ N &\rightarrow L \\ L &\rightarrow O L \\ L &\rightarrow O \\ O &\rightarrow 0 \mid \dots \mid 7 \end{aligned}$$

۲۲. برای پذیرش عبارات پسوندی با عملگرهای جمع و ضرب یک گراف نحو ارائه دهید. مثال: `id id * id +`. (۵نمره)

کد زیر را در نظر گرفته و به سوالات ۲۳ و ۲۴ پاسخ دهید.

```
1 public interface Infectious {
2     float R_nought();
3     void Spread();
4 }
5
6 public interface Cure {
7     string CureName();
8 }
9
10 public class abstract Disease {
11     public void getName();
12     public float FatalityRate();
13 }
14
15 public class Sars extends Disease implements Infectious {
16     private int numberOfInfectedCountries;
17
18     public void getName() {return "SARS";}
19     public float FatalityRate() {return 14.50; }
20
21     public float R_nought() {return 3;}
22     public void Spread() {numberOfInfectedCountries++;}
23 }
```

```

24     }
25
26     public class Sars_CoV_2 extends Sars implements Cure {
27         private int currentDeathInTheWorld;
28
29         public void getName() {return "Coronavirus";}
30         public float FatalityRate() {return 8.0;}
31
32         public string CureName() {return "Stay at home and build a compiler!";}
33
34         public void NewDeath() {currentDeathInTheWorld++;}
35     }

```

۲۳. جدول نمادها را به شیوهی پشت‌پشتی اسپاگتی^{۱۶} رسم کنید. (۳نمره)

۲۴. فرض کنید که در تابع main داریم: `Cure cure = new Sars_CoV_2()`. با توجه به این خط `vTable` را رسم کنید. (۳نمره)

۲۵. فرض کنید قصد داریم ساختار چندتایی مرتب^{۱۷} را به زبان Decaf اضافه کنیم. هر چندتایی مرتب می‌تواند به شکل زیر تعریف می‌شود: (۴نمره)

`Tuple(type 1, ..., type N)t = (const_type1, ..., const_typeN)`

به عنوان مثال یک دوتایی از اعداد صحیح و حقیقی به شکل زیر تعریف می‌شود

`Tuple(int, double)couple = (1, 3.14);`

به عنصر i ام یک چندتایی مرتب به صورت `t.i` دسترسی وجود دارد. همچنین این مورد هم برای انواع پایهای و هم برای انواع غیر پایهای قابل تعریف است. جمع میان چندتایی‌های مرتب که هم اندازه هستند و بین آن دو عمل جمع تعریف شده است ممکن است. در اینجا فرض کنید که عمل جمع می‌تواند برای انواع غیرپایهای نیز در صورت لزوم تعریف شود (در واقع ممکن است تعریف شده باشد و اگر تعریف شده باشد حاصل جمع برابر نوع `max` است و در غیر اینصورت `ErrorType`). سیستم نوع برای قاعده‌ی جمع را بنویسید.

۲۶. تابع `cegn` را برای دستور `if` در زبان Decaf بنویسید (متناظر با واژه‌ی نحوی `IfStmt`). (۴نمره)

بر اساس کد زیر به سوالات ۲۷ و ۲۸ جواب دهید.

```

1     a = 1 + 2
2     b = a + b
3     c = b + e
4     d = c + b
5     x = b + 3
6     t = c - 2
7     f = x + f
8     e = t - f
9     y = x - 2
10    d = d - y

```

۲۷. برای هر خط متغیرهایی که قبل از آن خط زنده هستند را بنویسید. فرض کنید که در نهایت `x` و `d` زنده‌اند. (۲نمره)

۲۸. فرض کنید که می‌توانید از هر یک از بهینه‌سازی‌هایی که در کلاس مطرح شد استفاده کنید. حداکثر چند بار اجرای انقیاد ثابت^{۱۸}، باعث بهینه‌تر شدن کد می‌شود؟ (۱نمره)

بر اساس کد زیر به سوالات ۲۹، ۳۰ و ۳۱ جواب دهید.

Spaghetti Stack^{۱۶}
 Tuple^{۱۷}
 Constant Folding^{۱۸}

```

1  if (condition1)
2  {
3      a = b + c;
4      c = d - 1;
5  }
6  else
7  {
8      while(condition2)
9      {
10         arr[i]++;
11         b = d + f;
12         if(condition3)
13         {
14             d = a + b - f;
15             e = c - d;
16             g = b + 2;
17         }
18         else
19         {
20             a = b + c;
21             f(a);
22         }
23         f = e + a;
24         c = a + b;
25     }
26 }
27 a = b + 2;
28 a = e + a;

```

(۱نمره)

۲۹. اگر بعد از این کد، بدانیم که d و c زنده خواهند بود لیست متغیرهای زنده در خط ۷ را بنویسید.

(۲نمره)

۳۰. بهینه‌سازی حذف افزونگی جزئی را در هر یک از خطوط ۲۷ و ۲۸ بررسی کنید.

(۱نمره)

۳۱. در چه صورت می‌توان خط ۱۰ را به خارج از حلقه منتقل کرد.

بر اساس کد زیر به سوالات ۳۲، ۳۳ و ۳۴ پاسخ دهید.

```

1  a = b - d
2  e = c + b
3  c = e - b
4  d = a + b
5  f = e
6  c = e + a
7  a = c + f
8  b = a
9  f = a
10 c = f + b
11 print(c)
12 print(a)

```

(۲نمره)

۳۲. گراف تصادم رجیسترها را برای کد فوق رسم کنید (نوشتن یال‌های کفایت می‌کند).

(۲نمره)

۳۳. کمترین تعداد رجیستر مورد نیاز که به کمک الگوریتم Chaitin بدست می‌آید چقدر است؟

(۱نمره)

۳۴. چه تغییری در گراف بدهیم که هم معتبر باشد و هم بتوان با یک رنگ کمتر آن را رنگ‌آمیزی کرد؟