



۱- درستی هر گزاره‌ی زیر را با دلیل مشخص کنید. بدون توضیح خوب، نمره‌ای به جواب یک قسمت تخصیص نمی‌یابد.

۱.۱ دو بلوک پایه‌ی متوالی می‌توانند در یک دستور اشتراک داشته باشند.

۲.۱ هر گرامری که توسط الگوریتم LL(1) قابل تجزیه باشد، توسط LALR(1) نیز قابل تجزیه است.

۳.۱ فرض کنید تابع func در عمق سه تعریف شده باشد (این تابع در تابعی تعریف شده است که خود در تابع دیگری تعریف شده است و آن تابع نیز در تابع دیگری تعریف شده است). برای دسترسی از تابع func به متغیری که در عمق یک تعریف شده است باید دو بار از لینک کنترل (Control Link) استفاده کرد.

۴.۱ تابع g در گراف فراخوانی، برگ است و رجیسترهای r1 و r2 را تغییر می‌دهد. اگر در این تابع فقط رجیستر r1 در پشته ذخیره شود، r2 قطعاً از نوع Caller-save است.

۵.۱ فرض کنید کلاس X شامل دو فیلد parent و child باشد. در شبه کد زیر و پس از اجرای این چهار دستور، الگوریتم شمارش ارجاع یک زباله و الگوریتم مبتنی بر Tracing (علامت‌گذاری و جاروب) دو زباله تشخیص می‌دهند.

```
a = X()
a.child = X()
a.parent = a
a = X()
```

۶.۱ برای متغیرهای گرامری مقدار مفهومی val تعریف شده است. قاعده‌ی زیر از این گرامر را در نظر بگیرید. فرض کنید عمل مفهومی این قاعده، پس از توکن to و قبل از متغیر Y و در سایر قواعد، عملیات مفهومی در پایان آنها قرار داشته باشند. عملیات مفهومی این گرامر در هنگام تجزیه توسط الگوریتم LL(1) قابل اجرا هستند.

$$X \rightarrow Y \text{ to } Z$$

۷.۱ قاعده‌ی نهم از گرامری به صورت زیر است. پس از محاسبه‌ی جدول تجزیه‌ی الگوریتم SLR، در یکی از خانه‌های جدول دو عمل r9 و s5 قرار گرفته‌اند. این خانه مربوط به سطری است که حالت معادل آن شامل Itemهای « $E \rightarrow E \circ E$ » و « $E \rightarrow E \circ E$ » است. برای اینکه عملگر \circ از راست شرکت‌پذیر باشد، باید r9 را از این خانه حذف کرد.

$$E \rightarrow E \circ E$$

۲- برای گرامر زیر فقط سطر مربوط به متغیر A در جدول LL(1) را نشان دهید (سایر اطلاعات لازم نیستند).

```
S → A print
A → B C
B → round
B → ε
C → circle
C → ε
```



- ۱۵ -۳- زبانی با گرامر زیر را در نظر بگیرید که حلقه‌ی do-while دارد (پس از یک بار اجرای بدنه‌ی حلقه، تا وقتی که شرط برقرار باشد بدنه‌ی حلقه تکرار می‌شود). به کمک عملیات و مقادیر مفهومی، برای این زبان کد میانی سه آدرس تولید کنید.

```

1 S → do { S } while ( E ) ;
2 S → id = E ;
3 E → num
4 E → id

```

- ۲۱ -۴- با توجه به کد میانی سه-آدرسه‌ی زیر، پاسخ دهید.

```

1 y = 5
2 x = 1
3 x = x * y
4 if x < y goto 3
5 return x

```

۱.۴ بلوک‌های پایه را مشخص کنید.

۲.۴ شکل SSA (Static Single Assignment Form) کد میانی را نمایش دهید.

۳.۴ نشان دهید چند بار و در کجای کد می‌توان از بهینه‌سازی‌های انتشار ثوابت

(Constant Folding) و عبارت مشترک (Common Subexpression) استفاده کرد.

- ۱۰ -۵- تحلیل زنده بودن (Liveness Analysis) را با نمایش گام‌ها برای کد میانی سؤال قبل انجام دهید.

با آرزوی موفقیت شما در این آزمون، درخواست می‌کنم به زمان پاسخگویی امتحان دقت کنید. همچنین، در زمان آزمون به پرسشی پاسخ داده نمی‌شود.