تمرین سری اول درس کامپایلر

توضیحات

سوالات تیوری را به صورت تایپ شده یا عکس دست نویس خوانا و مرتب آماده کنید.
پاسخ سوالات عملی را که هر یک در فایل‌های جداگانه (با نام سوال) قابل کامپایل و اجرا می‌باشد در یک فولدر قرار داده و در کنار فایل پاسخ سوالات تیوری با نام Name_Lastname_StudentNumber_HW1 فشرده سازی کنید و سپس در سامانه آپلود نمایید .
در مورد تکلیف میتوانید با دوستان خود مشورت کنید ولی تکلیف باید کار خود شما باشد.
در صورت کشف تقلب و موارد کپی، از هر دو شخص حداقل به میزان ۲۰ درصد نمره کسب شده، کسر خواهد شد.

۱- انواع خطاهایی که در زمان تحلیل لغوی می‌توان پیدا کرد را بنویسید. (غیر از آنچه در کلاس آموختید سرچ کنید فکر کنید و مواردی که به ذهنتان میرسد یا پیدا میکنید را دقیق بیان کنید) سپس برای هر یک یا هر دسته بنویسید چه روشی میتوان برای کشف و گذر از خطا پیشنهاد داد؟

۲- کامپایلرهای تک گذره(One pass) و چندگذره(Multi pass) را با یکدیگر مقایسه کنید.

۳- تفاوت اصلی کد میانی و کد نهایی تولید شده در طی مراحل یک کامپایلر را بیان کرده و مزیت ترجمه به کد میانی پیش از ترجمه به کد ماشین را توضیح دهید.

۴- در برنامه زیر، تعداد توکن‌ها و نوع آن‌ها را مشخص نمایید و قطعه کد را به دنباله‌ای از توکن‌ها تبدیل کنید.

```
int main(){
    // 2 variables
    int a = 10, b = 20;
    printf("sum is :%d", a+b);
    printf("i = %d, &i = %x", i, &i);
    return 0;
}
```

۵- برای هر یک از موارد زیر عبارت منظم مناسب را بنویسید.

الف) شماره تلفن:

(650)-723-3232

ب) آدرس ایمیل:

anyone@ec.iut.ac.ir

ج) یک عدد اعشاری(نمایش علمی)

د) فاصله‌ها

۶- برای عبارت منظم زیر NFA معادل رسم کنید و سپس به DFA معادل تبدیل کنید.

$$\left(((\mathbf{00})^*(\mathbf{11})) + \mathbf{01}\right)^*$$

۷- هر یک از خطاهای زیر در زبان C در کدام فاز کامپایلر تشخیص داده می‌شود؟

الف) برابرنبودن تعداد پرانتزهای باز و بسته در یک عبارت محاسباتی
ب) یکسان نبودن نوع متغیرهای a و b در عبارت a+b
ج) استفاده از واژه els به جای else
د) قرار دادن عبارت a=3 در داخل شرط if به جای a==3
ه) عدم تعریف پروتوتایپ یک تابع پیش از فراخوانی آن

۸- توکنهای A و B و C به صورت زیر مشخص شده‌اند. فرض کنید استراتژی longest match و سپس با اولویت ترتیب تعریف برای مشخص‌کردن نوع توکن استفاده می‌شود. خروجی رشته‌های ورودی داده شده چیست؟

TOKEN_A	cd*a*
TOKEN_B	c*a*cd
TOKEN_C	c*b

سوالات عملی

۱- به کمک کتابخانه RE در زبان پایتون عبارات منظم زیر را پیاده‌سازی کنید.
الف) تمامی رشته های باینری که عدد ۱ دقیقا چهار بار در آنها تکرار شده است
ب) تمامی اعداد دودویی که بزگتر از ۱۰۱۰۰۱ هستند
ج) تمامی اعداد دودویی که مضربی از ۵ هستند
د) تمام رشته‌هایی که حداکثر دو بار شامل زیررشته ۱۱ باشند

۲- برنامه lexای بنویسید که یک فایل cpp. دریافت کند و تمامی کامنت های یک خطی یا چند خطی آن را حذف کند، و سپس در یک فایل جدید آن را ذخیره کند.

۳- همانطور که میدانید در Lex، اگر از نماد r1/r2 استفاده شود (که r1 و r2 هر کدام یک regular expression هستند)، متن سمت چپ فقط در صورتی با r1 تطبیق پیدا می‌کند، که به دنبال آن متنی باشد که با r2 تطبیق پیدا کند. علاوه براین Lex می‌تواند متن سمت چپ را با کمک stateها تطبیق دهد. کد state_matching.lex که همراه با فایل تکلیف است، سعی می‌کند از این امکان استفاده کند. این کد را به صورتی کامل کنید که برای ورودیی شبیه زیر،

inputfile “fileA”
outputfile “fileB”

خروجی به صورت زیر باشد:

“fileA” is the input file”
“fileB” is the output file

۴- **(اختیاری و امتیازی، برای پاسخگویی به این سوال تا آخر ترم فرصت دارید پاسخ این سوال را در تکلیف جداگانه ای که در سامانه بدین منظور قرار گرفته آپلود کنید)** یک تحلیلگر لغوی برای شناخت توکن‌های زبان برنامه نویسی C++ بسازید به صورتی که از ساخت جدول transition برای تحلیل لغوی استفاده کند (از مجموعه ifelseها یا caseها استفاده نشود). برای شناسه ها و کلمات کلیدی (در صورت نیاز) جدول نمادها را طراحی و پیاده سازی کنید. در پایان اجرای برنامه باید جدول نمادها و دنباله توکن ها چاپ شود.