

# تمرین سری پنجم درس کامپایلر

بهار ۱۴۰۱

## توضیحات

سوالات تئوری 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 تئوری با نام Name\_Lastname\_StudentNumber\_HW5 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.

ا برای گرامر زیر و ورودی داده‌شده، درخت تجزیه را رسم کنید. روی درخت یالهای گراف dependency را مشخص کنید سپس دو topological sort مختلف از گراف بنویسید (نودهای گراف dependency را نامگذاری کرده و سپس با ذکر ترتیبی از نام نودهای گراف، topological sort را مشخص کنید.  
Integer x1,x2,x3

ب annotated tree (درختی که مقادیر ویژگیها روی نودهای آن بدست آمده) را برای ورودی Integer x1,x2,x3 طبق semantic ruleها رسم کنید.

ج این SDD را به SDT تبدیل کرده و آن را به روش اجرا بعد از ساخت درخت تجزیه از طریق پیمایش درخت، پیاده سازی کنید.

Production	Semantic rules
$D \rightarrow T L$	$L.a = T.type$
$T \rightarrow int$	$T.type = integer$
$T \rightarrow float$	$T.type = float$
$L \rightarrow L1, id$	$L1.a = L.a$ $addType(id.entry, L.a)$
$L \rightarrow id$	$addType(id.entry, L.a)$

۲. SDD زیر برای محاسبه مقدار عبارات ریاضی نوشته شده است. پیاده سازی این SDD را به دو روش bottom-up و top-down بنویسید (میتوانید آن را تبدیل به SDT کرده و سپس پیاده سازی کنید)

PRODUCTION	SEMANTIC RULES
1) $T \rightarrow F T'$	$T'.inh = F.val$ $T.val = T'.syn$
2) $T' \rightarrow * F T'_1$	$T'_1.inh = T'.inh \times F.val$ $T'.syn = T'_1.syn$
3) $T' \rightarrow \epsilon$	$T'.syn = T'.inh$
4) $F \rightarrow \mathbf{digit}$	$F.val = \mathbf{digit.lexval}$

۳. گرامر زیر رشته‌هایی تولید می‌کند که بیانگر اعداد باینری با امکان اعشاری بودن هستند (برای مثال ۱۰۱.۰۰۱۱ که دسیمال معادل آن ۵.۱۸۷۵ است). یک L-attributed SDD برای این گرامر بنویسید که با استفاده از حداقل یک ویژگی موروثی، مقدار دسیمال رشته ورودی را بدست آورد (دقت کنید که روش بدست آوردن دسیمال سمت چپ ممیز با سمت راست ممیز متفاوت است).

$S \rightarrow L.L \mid L$

$L \rightarrow LB \mid B$

$B \rightarrow 0 \mid 1$

۴. برای گرامری با قانون‌های زیر یک L-attributed SSD شبیه به مثال ۱۹ فصل ۵ کتاب مرجع (Example 5.19) بنویسید که هر کدام یک ساختار flow-of-control همانند زبان برنامه نویسی C نمایش می‌دهند. ممکن است به کدهای سه آدرسی برای پریدن (jump) به یک برچسب (label) خاص نیاز پیدا کنید، در هر مورد که چنین اتفاقی بیافتد باید goto L را تولید کنید.

ا

$S \rightarrow \mathbf{if} ( C ) S_1 \mathbf{else} S_2$

ب

$S \rightarrow \mathbf{do} S_1 \mathbf{while} ( C )$

ج

$S \rightarrow \{ ' L ' \}; L \rightarrow L S \mid \epsilon$

توجه داشته باشید که هر عبارت در لیست می‌تواند یک جهش از وسط خود به عبارت بعدی داشته باشد، بنابراین تولید کد برای هر عبارت به ترتیب کافی نیست.

۵. با اضافه کردن مارکرهای موردنیاز و با کمک روش backpatching، برای گرامر زیر actionهای لازم جهت تولید کد را بنویسید. (از توابع makelist، merge و backpatch استفاده کنید)

$S \rightarrow \text{If } (B) S1$

$B \rightarrow B1 \mid B2$

$B \rightarrow B1 \ \&\& \ B2$

$B \rightarrow \text{true}$

$B \rightarrow \text{false}$

۶. طرح ترجمه زیر را در نظر بگیرید.

$S \rightarrow ER$

$R \rightarrow *E\{\text{print}("*");\}R \mid \epsilon$

$E \rightarrow F + E \{\text{print}("+");\} \mid F$

$F \rightarrow (S) \mid \text{id} \{\text{print}(\text{id.value});\}$

در اینجا id یک نشانه است که یک عدد صحیح را نشان می دهد و id.value نشان دهنده مقدار صحیح مربوطه است.

برای رشته ورودی داده شده چه طرح ترجمه ای چاپ می شود؟  $2 * 3 + 4$

۷. طرح SDT را زیر را در نظر بگیرید.

$E \rightarrow E + E \quad \{\text{print} "+"\}$

$E \rightarrow E * E \quad \{\text{print} "."\}$

$E \rightarrow \text{id} \quad \{\text{print id.name}\}$

$E \rightarrow ( E )$

یک تجزیه کننده LR اعمال (actions) مرتبط با قوانین تولید (production rules) را بلافاصله پس از کاهش توسط قانون تولید مربوطه (production rule) مربوطه اجرا می کند. درخت تجزیه را رسم کنید و ترجمه جمله  $(a+b)*(c+d)$  را بنویسید.