ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

УПРАВЛЕНИЕ ДИСТАНЦИОННОГО ОБУЧЕНИЯ И ПОВЫШЕНИЯ КВАЛИФИКАЦИИ

Кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»

Вывод фразы простейшего языка и проверка нашего компилятора

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по специальностям
230105-«Программное обеспечение вычислительной техники и автоматизированных систем»
010503-«Математическое обеспечение и администрирование информационных систем»

Автор

Коледов Леонид Викторович

Ростов-на-Дону, 2013

Аннотация

Данная разработка может быть использована в качестве основного учебного материала по дисциплинам: «Теория языков программирования и методы трансляции» и «Теория вычислительных процессов»

Автор

Коледов Леонид Викторович, к. ф.-м. н., доцент, профессор кафедры

Область научных интересов

Информационные технологии, системы искусственного интеллекта

Оглавление

ЛАБОРАТОРНАЯ РАБОТА 2

Задание 1.

Используя материалы приложения 1, построить дерево синтаксического разбора для приведенных в таблице примеров.

Если выражение некорректно, укажите ошибку и поправьте ее.

Номер варианта	Задание 1	Задание 2
1	1+3/2*(6-3)-2-4-(4+7)	2-((1+3*2*(6-3)-2-4*(4+7))
2	4+2/3/2*(6+3)-2*4-(4-7)	2-(6+3*2/((6-3)-2)-4*(2+7)))
3	(1+3/2))*(6-3)-2-4-(4+7)	2-(1+3*2*(6-3))-2/(4*(4+7))
4	4+2/3/2*(2/6+3)-2/4+(4-7)	4*(-(9+3*2/((6-3)*5)-4/2+7)
5	1+3/2*(6-3)-2-4-(4+7)	2-((1+3*2*(6-3)-2-4*(4+7))
6	4+2/3*2-(6+3)-2*4-(4-7)	2-(6+3*2/((6-3)-2)-4*(2+7)))
7	(1+3*2)-((6-3)-2+4))-(1-4)+7	2-(1/3+2*(6-3))-2/(4*(6+7))
8	4+2/3/2*(2/6+3)-2/4+(4-7)	4*(-(9+3*2/((6-3)*5)-4/2+7)
9	1+3/2*(6-3)-2-4-(4+7)	2-((1+3*2*(6-3)-2-4*(3+7))
10	4+2/3/2*(6+3)-2*4-(4-7)	2-(6+3*2/((6-3)-2)-4*(2+7)))
11	(1+3/2))*(6-3)-2-4-(4+7)	2-(1+3*2*(6-3))-2/(4*(4+5))

12	4+2/3/2*(2/6+3)-2/4+(4-7)	4*(-(9+3*2/((6-3)*5)-4/2+7)
13	1+3/2*(6-3)-2-4-(4+7)	2-((1+3*2*(6-3)-2-4*(4+8))
14	4+2/3*2-(6+3)-2*4-(4-7)	2-(6+3*2/((6-3)-2)-4*(2+7)))
15	(1+3*2)-((6-3)-2+4))-(1-4)+7	2-(1/3+2*(6-3))-2/(4*(4+2))
16	4+2/3/2*(2/6+3)-2/4+(4-7)	4*(-(9+3*2/((6-3)*5)-4/2+3)
17	3/2/(2+4-3)*(2-7) 2/4+(4-7)	(1+3*2)-((6-3)-2+4))-(1-4)+7
18	4+2/3/2*(2/6+3)-2/4+(4-7)	4*(-(9+3*2/((6-3)*5)-4/2+7)
19	1+3/2*(6-3)-2-4-(4+7)	2-((1+3*2*(6-3)-2-4*(3+7))
20	4+2/3/2*(6+3)-2*4-(4-7)	2-(6+3*2/((6-3)-2)-4*(2+7)))
21	(1+3/2))*(6-3)-2-4-(4+7)	2-(1+3*2*(6-3))-2/(4*(4+5))
22	4+2/3/2*(2/6+3)-2/4+(4-7)	4*(-(9+3*2/((6-3)*5)-4/2+7)
23	1+3/2*(6-3)-2-4-(4+7)	2-((1+3*2*(6-3)-2-4*(4+8))
24	4+2/3*2-(6+3)-2*4-(4-7)	2-(6+3*2/((6-3)-2)-4*(2+7)))
25	(1+3*2)-((6-3)-2+4))-(1-4)+7	2-(1/3+2*(6-3))-2/(4*(4+2))

Задание 2.

Используя материалы лабораторной работы 1, построить трассировку решенного в предыдущем задании упражнения.

Проведите пошаговое исполнение программы с входными

файлами, куда включите и примеры с ошибками, а в отчет включите выходные файлы и поправки. По результатам исполнения постройте дерево вывода, соответствующего Вашему примеру.

Задание 3.

Постройте контекстно-свободную грамматику для римских чисел.

Приложение 1.

start	→	*list* **eof**	
list	→	*expr* ; *list*	
	|	ε	
expr	→	*expr* + *term*	{ *print*('+') }
	|	*expr* + *term*	{ *print*('-') }
	|	*term*	
term	→	*term* * *factor*	{ *print*('*') }
	|	*term* / *factor*	{ *print*('/') }
	|	*term* **div** *factor*	{ *print*('DIV') }
	|	*term* **mod** *factor*	{ *print*('MOD') }
	|	*factor*	
factor	→	(*expr*)	
	|	**id**	{ *print*(**id**.*lexeme*) }
	|	**num**	{ *print*(**num**.*value*) }

Рис. 2.35. Спецификация транслятора из инфиксной формы записи в постфиксную

Модуль синтаксического анализатора `parser.c`

start	→	*list* **eof**
list	→	*expr* ; *list*
	|	ε
expr	→	*term moreterms*
moreterms	→	+ *term* { *print*('+') } *moreterms*
	|	- *term* { *print*('-') } *moreterms*
	|	ε
term	→	*factor morefactors*
morefactors	→	* *factor* { *print*('*') } *morefactors*
	|	/ *factor* { *print*('/') } *morefactors*
	|	`div` *factor* { *print*(`'DIV'`) } *morefactors*
	|	`mod` *factor* { *print*(`'MOD'`) } *morefactors*
	|	ε
factor	→	(*expr*)
	|	**id** { *print*(**id**.*lexeme*) }
	|	**num** { *print*(**num**.*value*) }

Рис. 2.38. Схема синтаксически управляемой трансляции после устранения левой рекурсии