Отчет по лабораторной работе #2

Михаил Волхов (M3338)

28 апреля 2016 г.

1 Условия

Арифметические выражения в постфиксной записи с операциями сложения, вычитания, умножения. Операнды – целые числа.

2 Разработка грамматики

В первую очередь заметим, что арифметические выражения в постфиксной нотации не имеют встроенных в грамматику приоритетов операций – любое выражение может быть разобрано однозначно.

Поэтому изначальная грамматика будет выглядить так:

$$\begin{aligned} A &\to AA* \\ A &\to AA+ \\ A &\to AA- \\ A &\to n \end{aligned}$$

В нотации EBNF:

$$A \to AA* | AA+ | AA- | n$$

Избавимся сначала от левой рекурсии:

$$\begin{aligned} A &\to nB | n \\ B &\to A* | A+ | A- | A*B | A+B | A-B \end{aligned}$$

Есть правое ветвление, устраним его:

$$\begin{aligned} A &\to nB | n \\ B &\to AC \\ C &\to * | + | - | *B | +B | -B \end{aligned}$$

Все еще есть правое ветвление, устраним:

$$\begin{aligned} A &\to nB | n \\ B &\to AC \\ C &\to *D | +D | -D \\ D &\to \varepsilon | B \end{aligned}$$

И в первом правиле:

$$\begin{aligned} A &\to nD \\ B &\to AC \\ C &\to *D | +D | -D \\ D &\to \varepsilon | B \end{aligned}$$

3 Построение лексического анализатора

Лексический анализатор находится в модуле `Recparser.Lexer`. Терминалы в нашей грамматике: $n, +, -, *$. Соотвествуют дататайпу `Token` в представленном модуле.

4 Построение синтаксического анализатора

Посчитаем `FIRST` и `FOLLOW`:

T	FIRST	FOLLOW
A	n	$, *, +, -
B	n	$, *, +, -
C	*, +, -	$, *, +, -
D	ε, n	$, *, +, -

Код синтаксического анализатора находится в модуле `Recparser.Parser`. Дататайпы `PA, PB, PC, PD` в точности соответствуют нетерминалам грамматики.

5 Тестирование

Тестирование проводится с помощью `hspec`, тесты делятся на ручные и автоматически сгенерированные (`quickcheck`).

Автоматически сгенерированные тесты:

- Лексер принимает случайные правильные строки токенов (длина 200).
- Лексер отвергает случайные правильные строки токенов вперемешку с мусором (символы, не встречающиеся в грамматике, тут и далее).
- Парсер корректно работает на случайном дататайпе грамматики (x100, разная глубина).
- Парсер падает, получив на вход случайный дататайп грамматики (те же параметры) + мусор.

Ручные тесты:

Правило	Вердикт	ОБъяснение
1	Принимает	Правило A
-1	Принимает	То же самое
1 2 +	Принимает	Правило A + D + B
-1 -2 +	Принимает	То же самое
1 2 + 3 4 * -	Принимает	Более сложный тест на B
1 2 + 3 4 * 5 - *	Принимает	Два C (арифметических операции) подряд
1 2 + 5 3 4 * - *	Принимает	То же, но в другом порядке
a	Не принимает	Незнакомый символ для лексера
*	Не принимает	Правило A
1 +	Не принимает	A + суффикс
1 2 + *	Не принимает	A + B + D + суффикс
1 2 + 5	Не принимает	То же, другой суффикс (A вместо C)