

Упрощение деревьев выражений

Лексемы

Дерево выражений

Задание

Вам необходимо реализовать функцию, преобразовывающую входящее выражение, согласно выбранному варианту. Функция должна обрабатывать все вхождения сходных фрагментов в анализируемых выражениях, а не только те, что приведены в примерах.

Функция преобразования должна быть реализована в `transform.c` в виде статической функции и вызвана в одной из функций `simplify_expression` или `expand_expression`, если преобразование упрощает или раскрывает выражение соответственно.

Уточнения по реализации

Примеры реализации функции упрощения и раскрытия выражения можно найти в том же файле `transform.c`.

`API parser.h` для работы с выражениями, а также вспомогательные функции в `transform.c` достаточны для реализации функции преобразования.

Приветствуются свои вспомогательные функции для упрощения реализации преобразования.

Варианты

В примерах слева от стрелки приводится фрагмент выражения до, а справа – после выполнения преобразования.

1. Упростить выражения, выполнив сложение:

$$2 + 3 \rightarrow 5, (-2 + b + 4) * k \rightarrow (2 + b) * k.$$

2. Упростить выражения, выполнив вычитание:

$$3 - 5 \rightarrow -2, 5 + b - 3 \rightarrow 2 + b.$$

3. Упростить выражения, выполнив умножение:

$$2 * b * 2 \rightarrow 4 * b.$$

4. Упростить выражения, выполнив деление, если они делятся без остатка:

$$4 * a / 2 \rightarrow 2 * a.$$

5. Упростить выражения, выполнить возведение в степень с целым показателем:

$$2^3 \rightarrow 8.$$

6. Упростить выражения, выполнив приведение подобных членов.
7. Редуцировать выражения, заменив операцию умножения переменной на целое число n на сумму n слагаемых:

$$a * 3 \rightarrow a + a + a.$$

8. Редуцировать выражения, заменив операцию возведения переменной в целую степень n на произведение n слагаемых:

$$a^3 \rightarrow a * a * a.$$

9. Умножение переменной на сумму заменить на сумму произведений.
10. Умножение переменной на разность заменить на разность произведений.
11. Вынести общие сомножители (переменные и константы) из суммы.
12. Вынести общие сомножители (переменные и константы) из разности.
13. Упростить выражения, убрав из него все произведения, в которых в качестве сомножителя используется ноль.
14. Убрать из выражений все слагаемые, равные нулю.
15. Убрать из выражений все сомножители, равные единице.
16. Убрать из частных все делители, равные единице.
17. Перемножить степени с одинаковыми основаниями:

$$a^2 * a^k \rightarrow a^{(2+k)}.$$

18. Вынести из произведений унарные минусы:

$$a * -b * 3 \rightarrow -(a * b * 3), \quad a * -b * 4 * -5 \rightarrow a * b * 4 * 5.$$

19. Упростить сложные (многоэтажные) дроби:

$$(a/b)/c \rightarrow a/(b * c), \quad a/(b/c) \rightarrow (a/b) * c.$$

20. Заменить вычитание на сложение с противоположным числом:

$$a - b * c \rightarrow a + (-(b * c)).$$

21. Выполнить сложение и вычитание дробей:

$$a/b + c/d \rightarrow (a * d + b * c)/(b * d).$$

22. Упростить дробь, сократив в числителе и знаменателе общие переменные и константы:

$$(a * b * 3)/(3 * b * c) \rightarrow a/c, \quad a/(a * b) \rightarrow 1/b.$$

23. Перемножить дроби:

$$(a/b) * (c/d) \rightarrow (a * c)/(b * d).$$

24. Заменить степень с суммой в показателе на произведение степеней:

$$a^{b+c} \rightarrow a^b a^c.$$

25. Заменить степень с разностью в показателе на частное степеней:

$$a^{b-c} \rightarrow a^b / a^c.$$

26. Заменить дробь со степенью в знаменателе произведением на степень с отрицательным показателем:

$$a / (b^c) \rightarrow a * b^{-c}.$$

27. (???) Поменять местами операнды у операций сложения и умножения. Учитывать приоритет и ассоциативность операций.

28. Выполнить замену переменной на выражение:

$$a ::= i + 4, a + 2 * a \rightarrow i + 4 + 2 * (i + 4).$$

29. Упорядочить соседние операнды в операциях сложения и умножения, вынося вперед константы. Сохранить порядок следования констант и переменных:

$$a + 3 + b + a * c + 2 \rightarrow 3 + 2 + a + b + a * c.$$

30. Лексикографически упорядочить соседние операнды операции сложения и умножения:

$$2 + b + c + a - a * b \rightarrow 2 + a + b + c - a * b.$$

31. Разложить на множители квадрат суммы:

$$(a + b)^2 \rightarrow a^2 + 2 * a * b + b^2.$$

32. Свернуть множители в квадрат суммы.

33. Разложить на множители квадрат разности:

$$(a - b)^2 \rightarrow a^2 - 2 * a * b + b^2.$$

34. Свернуть множители в квадрат разности.

35. Разложить на множители сумму кубов:

$$a^3 + b^3 \rightarrow (a + b) * (a^2 - a * b + b^2).$$

36. Свернуть множители в сумму кубов.

37. Разложить на множители разность кубов:

$$a^3 - b^3 \rightarrow (a - b) * (a^2 + a * b + b^2).$$

38. Свернуть множители в разность кубов.

39. Разложить на множители квадратный трехчлен
 40. Преобразовать многочлен в стандартный вид, располагая слагаемые заданного многочлена по степеням x в порядке возрастания или убывания:

$$f(x) = \sum_{i=0}^n a_i x^i.$$

41. В многочлене

$$f(x) = \sum_{i=0}^n a_i x^i$$

выполнить замену a_i на число i и на выражение $a_i - i$.

42. Заданы два многочлена

$$f(x) = \sum_{i=0}^n a_i x^i, g(x) = \sum_{i=0}^n b_i x^i.$$

Построить новый многочлен

$$h(x) = \sum_{i=0}^n c_i x^i,$$

где $c_i = \max(a_i, b_i)$, $c_i = \min(a_i, b_i)$, $c_i = a_i \cdot b_i$ и $c_i = \frac{a_i}{b_i}$.

43. Дан многочлен $f(x) = \sum_{i=0}^n a_i x^i$. Построить многочлен $g(x) = \sum_{i=0}^n a_{n-i} x^i$.
 44. Умножить многочлен от x на число k .
 45. Поделить многочлен от x на число k .
 46. Умножить многочлен от x на переменную x .
 47. Поделить многочлен от x на переменную x .
 48. (***) Выполнить полное дифференцирование выражения.

Если задания сверху кажутся скучными, то можно сделать следующее: - Реализовать построение дерева выражений с помощью алгоритма Дейкстры (Shunting yard algorithm); - Реализовать поддержку распространенных математических функций (sin, cos, tan, abs, sign, ...); - Реализовать поддержку распространенных математических констант (pi, e, tau, phi, ...);

Свои предложения можно так же обсудить с админом.

Тестирование

В этой работе присутствует автоматическое тестирование.

Функцию преобразования необходимо проверить на нескольких выражениях, среди которых должны быть выражения, не содержащие преобразуемых элементов, содержащие ровно один такой элемент или несколько элементов, подлежащих преобразованию, причем в “разных” местах выражения.

Свои тесты следует написать в директории `tests`. В этой директории есть поддиректории `simplify` и `expand` для тестирования функций преобразования упрощения и раскрытия выражения соответственно.

Создайте директорию в одной из этих поддиректорий, имеющее то же имя, что и Ваша функция, но все пробелы и нижние подчеркивания следует заменить на дэш (-) и все буквы должны быть нижнего регистра.

Например, функция разложения на множители разности квадратов реализована под именем `factor_difference_of_squares`, она раскрывает выражение, поэтому тесты должны находиться в директории `factor-difference-of-squares` в директории `tests/expand`. То есть полный путь до директории будет `tests/expand/factor-difference-of-squares`.

Каждый тестовый файл в новой директории должен содержать `-test` в конце его имени, не содержать пробелов и содержать латинские буквы нижнего регистра. Вместо пробелов используйте дэш (-).

Каждому тестовому файлу должен быть сопоставлен файл с правильным ответом. В имени файла с ответом: `-test` должен быть заменен на `-answer`.

Например написаны 3 теста: `empty-test`, `simple-test`, `my-advanced-test`. Для них присутствуют правильные ответы `empty-answer`, `simple-answer`, `my-advanced-answer`.

Ваше решение будет проверяться скриптом:

```
./test.sh
```

Прежде чем делать Pull Request в оригинальный репозиторий, рекомендуется запустить скрипт `test.sh` и проверить, что все тесты проходятся успешно, иначе работа на Code Review проверяться не будет.

Очень рекомендуется собирать программу и запускать тестирующий скрипт локально, без применения для этого веб-сервисов и многочисленных пушей в репозиторий.

Важные требования к решению

Чтобы получить максимальный балл за задачу, необходимо удовлетворять следующим критериям: - Код должен быть понятным, названия переменных отражают то, что они хранят; - Запрещается менять основной API; - Обязательно придумать хорошие тесты, см. выше пункт Тестирование; - Запрещается использовать сторонние библиотеки, кроме стандартной;