



Описание задачи

- В первых пяти задачах требуется расширить синтаксис базисных рефал-программ: добавить собственные синтаксические конструкции, которые посредством исполнения промежуточного модуля будут транслироваться в базисные рефал-конструкции. На вход вашей программе должна подаваться программа в расширенном синтаксисе, результатом её работы — семантически эквивалентная ей программа в стандартном синтаксисе (компилируемая Рефалом-5).



Описание задачи

- В 6 варианте на вход может подаваться одна или две программы на базисном Рефале. Если подаётся только одна программа, требуется построить её граф зависимостей (в dot-формате, как пример). Если две — требуется проверить, совпадают ли их графы зависимостей без учёта мусорного кода и именования функций.
- Комментарии преобразовывать не надо ни в одной задаче (в том числе задаче 7).



Описание задачи

- Варианты 7 и 10 подразумевают определённый эвристический подход: следует изобрести соответствующие преобразования. При этом разрешается использовать внешние вызовы через System, но только других рефал-программ (например, автоматически порождённых).
- В варианте 8 санитайзер должен работать даже для нестатических вызовов System (через Mu, принимающий переменную в качестве имени функции).
- Статический стек — тот, к которому программа всегда обращается по его имени (а не через переменную). Если во входной программе задачи 9 есть обращения к стекам через переменные, необходимо сообщить об этом и завершить выполнение программы.

- ❶ (Вариант 1) Добавить в Рефал переменные типа строка символов и типа число;
- ❷ (Вариант 2) Добавить многоместные функции;
- ❸ (Вариант 3) Добавить возможность использовать отрицания в условиях (причём свежие аргументы отрицаний не могут входить в правые части правил);
- ❹ (Вариант 4) Добавить возможность вызывать функции высших порядков из модуля без объявления в этом модуле всех функций, являющимися их аргументами;
- ❺ (Вариант 5) Добавить возможность использовать альтернативу (дизъюнкцию) в условиях (с приоритетом выбора значения при дальнейшей подстановке в правую часть);

- ❶ (Вариант 6, +1 балл) Реализовать модуль, анализирующий графы зависимостей вызовов в базисных рефал-программах и проверяющий, совпадают ли они в двух программах без учёта мусорного кода;
- ❷ (Вариант 7, +1 балл) Реализовать обфускатор, преобразующий код на базисном Рефале в семантически эквивалентный ему, но имеющий другую структуру зависимостей вызовов;
- ❸ (Вариант 8, +2 балла) Реализовать локальный санитайзер, запрещающий вызывать функцию `System` (и не запрещающий вызывать другие стандартные функции, в том числе через `Mu`). Подразумеваем, что все функции — либо встроенные, либо объявлены в этом же модуле;
- ❹ (Вариант 9, +3 балла) Реализовать автоматическое извлечение в код статических стеков;
- ❺ (Вариант 10, до +5 баллов) Реализовать деобфускатор, находящий в базисных рефал-программах функции, которые возвращают на всех аргументах пустое слово (эвристическое решение задачи).