О верификации программ, манипулирующих строковыми данными

Антонина Николаевна Непейвода

Институт программных систем РАН

При вычислениях над параметризованными выражениями типа строка возникает задача анализа уравнений в словах.

Определение:

Пусть фиксированы некоторый конечный алфавит символов-констант A и алфавит строковых переменных X. Уравнение в словах есть выражение w_1=w_2, где w_1, w_2 - слова в алфавите A u X (объединении алфавита символов и алфавита строковых переменных). Решить уравнение в словах - значит найти все значения переменных, входящих в w_1 или w_2, такие, что после подстановки этих значений вместо соответствующих переменных в уравнение w_1 = w_2 это уравнение превращается в графическое равенство.

Г.С. Маканин (1977 г.) описал нетривиальный алгоритм, перечисляющий все решения данного уравнения в словах с n переменными. Алгоритм Маканина является очень сложным. Он в течение долгого времени интенсивно изучался многими авторами. В последнее время (2014-2015) был сделан значительный шаг в понимании природы задачи. A. Jez (2014-2015 гг.) предложил недетерминированный алгоритм решения уравнений в словах, основанный на оригинальных идеях. W. Plandowski изучал класс множеств, элементы которого являются множествами решений конкретных уравнений в словах. Данный класс множеств не включает в себя и не включается в множество автоматно распознаваемых слов и отношений.

Уравнения в словах могут появляться непосредственно при развертке дерева программы над строковым типом на параметризованных входных данных.

Пример:

Рассмотрим следующую функцию.

```
F( string x, string y ) {

      if (x == y) then { … }
```

```
    else { … }

}
```

Предположим, что нужно анализировать поведение этой программы в контексте параметризованного вызова F(T(p1,..,pN), S(p1,..,pN)), где T(p1,..,pN) и S(p1,..,pN) суть выражения, параметризованные строковыми параметрами p1,..,pN.

Тогда возникает следующее уравнение в словах

T(p1,..,pN) = S(p1,..,pN),

от свойств решений которого зависит поведение анализируемой программы.

Уравнения в словах также могут быть введены алгоритмом преобразования (или анализа) программы, манипулирующей строками символов, перед которым стоит эта задача. Поскольку множества строк, являющихся решениями уравнений в словах, в общем случае не выразимы в терминах простых (контекстно-свободных) грамматик, уравнения в словах также можно использовать как компактное описание свойств допустимых значений строковых параметров, связанных этими уравнениями.

Пример:

Ниже бинарная инфиксная операция ++ есть ассоциативная операция приписывания над строковым типом.

Уравнение в словах x++y = y++x описывает следующее нетривиальное свойство. Строковые параметры x и y являются степенями некоторой неизвестной строки z, т.е. x = z++…++z (n раз) и y=z++...++z (m раз), где m и n - целые числа, большие или равные 1.

Суперкомпиляция – это метод преобразования программ, основанный на развёртке дерева параметризованных состояний программы и существенно использующий алгоритмы сопоставления параметризованных данных и их обобщения. В докладе будет рассмотрен метод использования уравнений в словах в качестве языка описания свойств допустимых значений строковых параметров, описывающих параметризованные состояния преобразуемой программы. Данный метод реализован в модельном суперкомпиляторе MSCP-A, оптимизирующем программы на языке программирования Рефал.