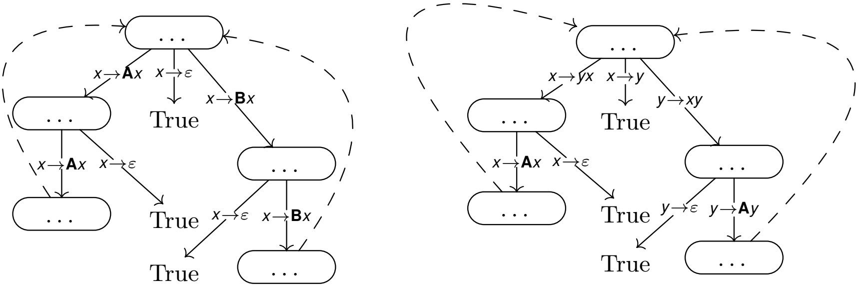
Применение уравнений в словах при преобразовании программ над строковым типом

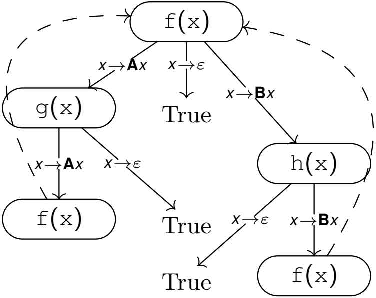
Антонина Непейвода

Семинар лаборатории автоматизации программирования,
29 ноября 2017.

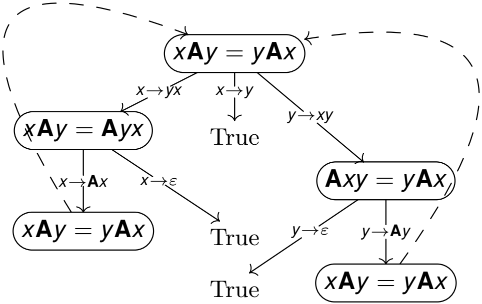
Что изображено на графах?



Что изображено на графах?



Развертка графа вычислений программы, распознающей язык $\{\mathbf{AA}|\mathbf{BB}\}^*\{\mathbf{A}|\mathbf{B}|\varepsilon\}$.



Граф решения уравнения в словах с множеством решений $x \in \{w\mathbf{A}\}^* w$, $y \in \{w\mathbf{A}\}^* w$.

Псевдокод программ над строковым типом

- ε — пустая строка, : — операция приписывания строк;
- **A**, **B**, **C** и т.д. — буквы (элементы алфавита);
- u, v, w — параметры типа строка; t — параметр типа буква;
- $\mathtt{x}$, $\mathtt{y}$, $\mathtt{z}$ — переменные типа строка; $\mathtt{s}$ — переменная типа буква.

Определение функции $\mathtt{F}(\mathtt{x}_1, \dots, \mathtt{x}_n)$ — набор правил:

$$\begin{array}{rcl}
\mathtt{F}([\text{Pattern}]_1^1, \dots, [\text{Pattern}]_n^1) & = & [\text{Expression}]_1; \\
& \dots & \\
\mathtt{F}([\text{Pattern}]_1^k, \dots, [\text{Pattern}]_n^k) & = & [\text{Expression}]_k;
\end{array}$$

$[\text{Pattern}]_j^i$ (образцы) не содержат вызовов.

Подстановка выражения в образцы осуществляется последовательно начиная с первого сверху правила до первого успеха.

Пример

$$
\begin{array}{lcl}
\mathtt{BadF}(\mathtt{x},\mathtt{y},\mathtt{y}) & = & \mathtt{AllB}(\mathtt{x});\\
\mathtt{BadF}(\mathtt{z}_1,\mathtt{z}_2,\mathtt{z}_3) & = & \mathbf{F};\\
\\
\mathtt{AllB}(\varepsilon) & = & \mathbf{T};\\
\mathtt{AllB}(\mathbf{B}{:}\mathtt{x}) & = & \mathtt{AllB}(\mathtt{x});\\
\mathtt{AllB}(\mathsf{s}{:}\mathtt{z}) & = & \mathbf{F};
\end{array}
$$

При вызове $\mathtt{AllB}(w)$ третье правило будет применено в том и лишь в том случае, если первая буква w — не $\mathbf{B}$.

Первое правило `BadF` содержит повторные вхождения переменных. При попытке подстановки в его образец параметризованного выражения (т.н. прогонке) возникнет уравнение в словах.

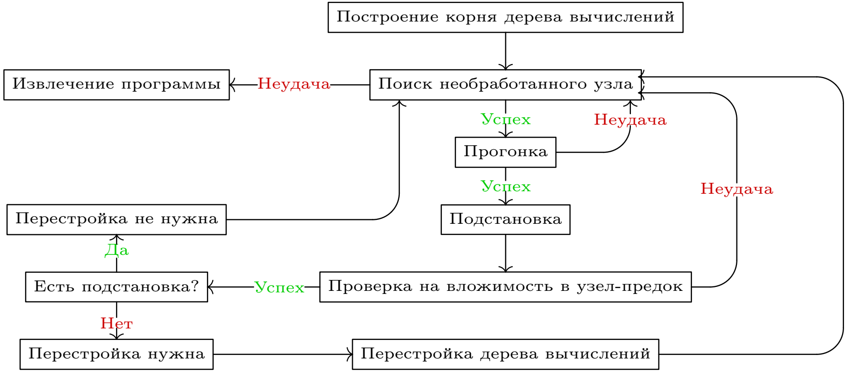
При каких значениях w, v вызов $\mathtt{BadF}(v, w{:}w{:}w, v{:}\mathbf{A}{:}v)$ возвращает $\mathbf{T}$?

Уравнения как язык описания свойств параметров

- Уравнения в словах — короткий способ записи сложных свойств параметров.
  - $uv = wu \Rightarrow \exists u_1, u_2, n(u = (u_1 u_2)^n u_1 \& v = u_2 u_1 \& w = u_1 u_2)$.
- Иногда — единственный.
  - (Хмелевский, 1971): решения уравнения $uw_1 v = vw_2 u$ нельзя переписать как конечное множество приписываний строк и строковых параметров, возведенных в переменные степени.

Даже имея полностью завершенный граф решения уравнения, неясно, как извлечь из него решения в явном виде.

Суперкомпиляция выражения $\mathtt{BadF}(v, w{:}w{:}w, v{:}\mathbf{A}{:}v)$



Корень дерева вычислений содержит выражение $\mathtt{BadF}(v, w{:}w{:}w, v{:}\mathbf{A}{:}v)$.
Попытка сопоставления с образцом сразу выделяет уравнение
$www = v\mathbf{A}v$.

Что делать с уравнениями? (Karhumäki, 2001)

- Расщеплять.
  - $u\mathbf{A}v = \mathbf{A}u\mathbf{B}uu \iff u\mathbf{A} = \mathbf{A}u$ и $v = \mathbf{B}uu$.
  - $u\mathbf{AB}u = vt_1w\mathbf{A}wv \iff u\mathbf{A} = vt_1w$ и $\mathbf{B}u = \mathbf{A}wv$.
- Оценивать длины.
  - $u\,t\,v = v\mathbf{A}t \Rightarrow$ длина u равна 1.
  - $wt_1wut_2w = v\mathbf{A}wvu \Rightarrow$ уравнение не имеет решений.
- Применять правила замены (лемму Леви):
  $u\Phi = v\Psi \Rightarrow$ либо $u = v$, либо $u = vu_1$, либо $v = uv_1$.
  - В общем случае приводит к неограниченному росту размеров уравнения.
- Решать в частных случаях:
  - квадратичные уравнения;
  - бескоэффициентные уравнения;
  - уравнения с одной неизвестной;
  - ...

Схема упрощения уравнения с разделяющимися переменными

Даны параметризованные строки $\Phi(v_1, \dots, v_n)$ и $\Psi(w_1, \dots, w_m)$, $w_i \neq v_i$.
Построить подстановки $\xi(v_1), \dots, \xi(v_n)$: $\Phi(\xi(v_1), \dots, \xi(v_n)) = \Psi(w_1, \dots, w_m)$.

Пусть $\Delta[1]$ есть первая буква или параметр строки Δ.

- $\Phi = \varepsilon$ или $\Psi = \varepsilon$ — сопоставление завершено.
- $\Phi[1]$ и $\Psi[1]$ — буквы (известные или неизвестные) — сопоставление $\Phi[1]$ и $\Psi[1]$ однозначно.
- $\Phi[1]$ — буква t, $\Psi[1] = w_i$. Расщепление:
  - $w_i = \varepsilon$;
  - $w_i = t\, w_i'$, где w_i' — новый параметр.
- $\Psi[1]$ — буква t, $\Phi[1] = v_i$. Расщепление:
  - $v_i = \varepsilon$;
  - $v_i = t\, v_i'$, где v_i' — новый параметр.
- $\Psi[1] = w_i$, $\Phi[1] = v_j$. Расщепление:
  - $v_j = \varepsilon$;
  - $w_i = w_i^1\, t\, w_i^2$ и $v_j = w_i^1\, t$;
  - $v_j = w_i v_j'$.

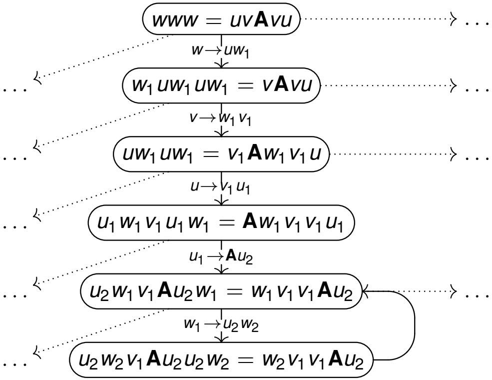
Особенности предложенного способа

- Осуществляется с двух сторон.
- Сужения параметров w_i подставляются в $\Psi(w_1, \dots, w_m)$ всюду; подстановки параметров в $\Phi(v_1, \dots, v_n)$ подставляются только в то вхождение, которое их породило. Поэтому шаг развертки уравнения укорачивает либо правую часть, либо левую часть уравнения.
- Возможны несколько подстановок в v_i, что порождает уравнения на $w_1, \dots, w_m$.

Сравнение с разверткой по лемме Леви

Односторонняя:

- асимметричная;
- всегда завершается;
- результат выписывается как конечное множество подстановок и уравнений.

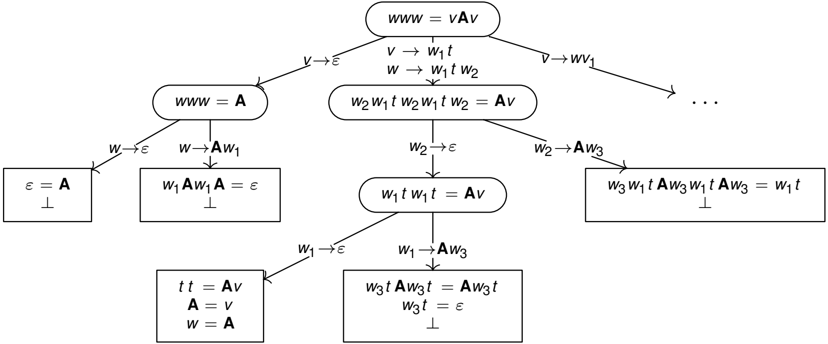


По лемме Леви:

- симметричная;
- на подходящих уравнениях быстрее определяет наличие корней;
- трудно извлекаемый результат.

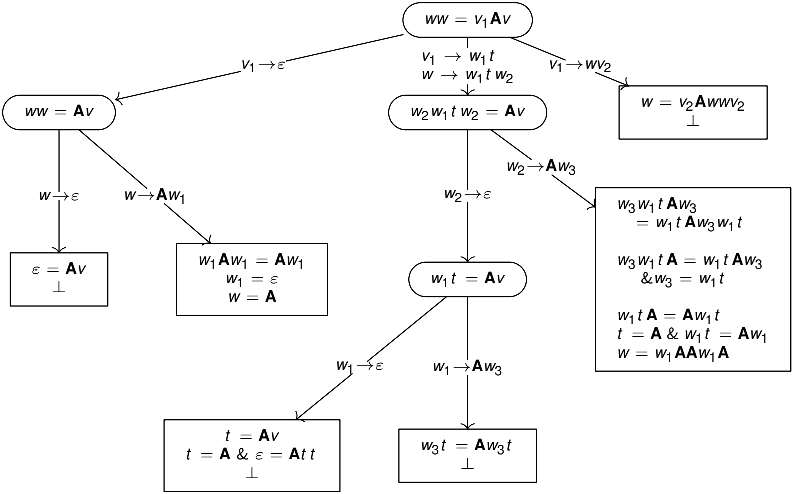
Развертка уравнения $www = v\mathbf{A}v$

Уравнение $www = v\mathbf{A}v$ решается относительно левой части.

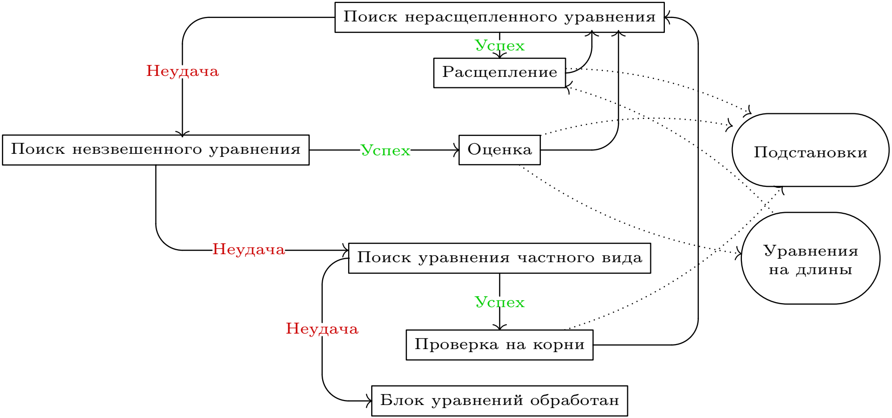


Развертка уравнения $www = v\mathbf{A}v$. Продолжение

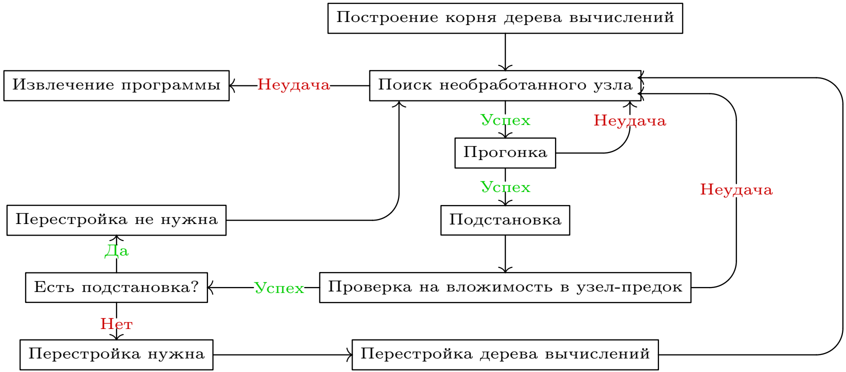
Развертка ветви, начинающейся от корня, по сужению $v \to wv_1$.



Обработка уравнений в MSCP-A



Суперкомпиляция выражения `BadF`(v, www, $v\mathbf{A}v$)



- $w = \mathbf{A}$, $v = \mathbf{A}$ — вызывается `AllB`($\mathbf{A}$), возвращает $\mathbf{F}$.
- $w = w_1\mathbf{AA}w_1\mathbf{A}$, $v = w_1\mathbf{AA}w_1\mathbf{A}w_1\mathbf{A}$ и $w_1\mathbf{A} = \mathbf{A}w_1$ — вызывается `AllB`($w_1\mathbf{AA}w_1\mathbf{A}w_1\mathbf{A}$), возвращает $\mathbf{F}$.
- В других случаях нет решений уравнения $v\mathbf{A}v = www$.

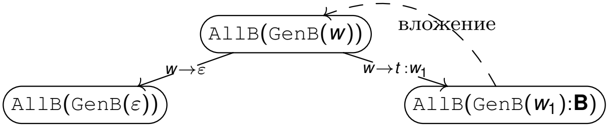
Пример взаимодействия с обобщением

$$\begin{array}{lcl}
\texttt{GenB}(\varepsilon) & = & \varepsilon; \\
\texttt{GenB}(\texttt{s}{:}\texttt{x}) & = & \texttt{GenB}(\texttt{x}){:}\mathbf{B}; \\
\\
\texttt{AllB}(\varepsilon) & = & \mathbf{T}; \\
\texttt{AllB}(\mathbf{B}{:}\texttt{x}) & = & \texttt{AllB}(\texttt{x}); \\
\texttt{AllB}(\texttt{s}{:}\texttt{z}) & = & \mathbf{F};
\end{array}$$

Функция `GenB` порождает строку из букв **B** с конца, а функция `AllB` просматривает ее сначала.

Что MSCP-A скажет про возможные результаты вызова `AllB(GenB(`***w***`))`?

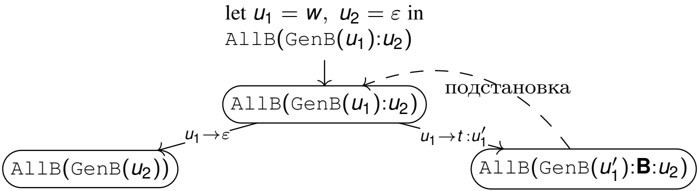
Суперкомпиляция выражения AllB(GenB(w))



Обобщение выражений AllB(GenB(w)) и AllB(GenB(w_1):**B**) — выражение $\Phi(u_1, \ldots, u_n)$ и подстановки ξ_1 и ξ_2 такие, что

- $\Phi(\xi_1(u_1), \ldots, \xi_1(u_n)) =$ AllB(GenB(w)),
- $\Phi(\xi_2(u_1), \ldots, \xi_2(u_n)) =$ AllB(GenB(w_1):**B**).

Пусть AllB(GenB(u_1):u_2), $\xi_1(u_1) = w$, $\xi_1(u_2) = \varepsilon$, $\xi_2(u_1) = w_1$, $\xi_2(u_2) = \mathbf{B}$.
Требуется перестройка дерева вычислений.

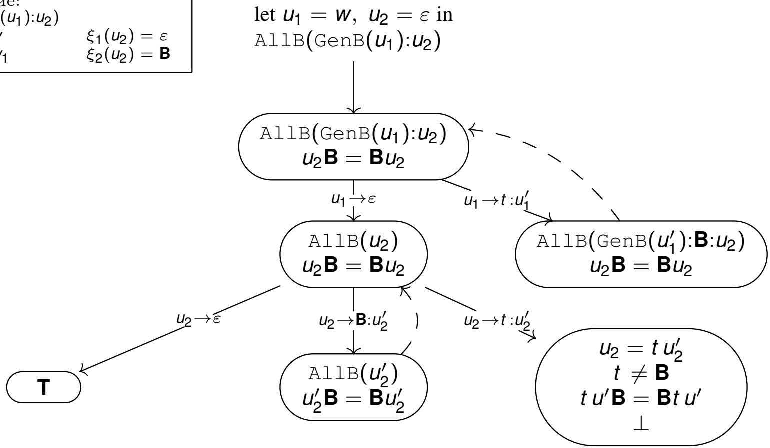


Суперкомпиляция выражения $\mathtt{AllB}(\mathtt{GenB}(w))$

Обобщение:
$\mathtt{AllB}(\mathtt{GenB}(u_1){:}u_2)$

$\xi_1(u_1) = w$ $\qquad$ $\xi_1(u_2) = \varepsilon$

$\xi_2(u_1) = w_1$ $\qquad$ $\xi_2(u_2) = \mathbf{B}$



Подстановки $\xi_1(u_2) = \varepsilon$ и $\xi_2(u_2) = \mathbf{B}$ удовлетворяют уравнению $u_2\mathbf{B} = \mathbf{B}u_2$. Это уравнение входит в противоречие с запретом $t \neq \mathbf{B}$ на ветке дерева вычислений, возвращающей $\mathbf{F}$.

Важно! При всех подстановках уравнение сохраняется!

Резюме

- Язык уравнения в словах — удобное средство выражения свойств параметризованных программ, оперирующих строками.
- Сопоставление параметризованных данных с образцом с повторными переменными — задача, необходимо обращающаяся к задаче упрощения уравнений в словах.
- Предложено обобщение, способное порождать уравнения – гипотезы о свойствах параметризованных состояний. Подстановки могут потверждать или опровергать эти гипотезы. Цена — дополнительные перестроения дерева вычислений.

Спасибо за внимание!