Применение уравнений в словах при преобразовании программ над строковым типом

Антонина Непейвода

Институт Программных Систем РАН

Семинар ИСП РАН «Технологии разработки и анализа программ»
1 марта 2018

Строковый тип данных и уравнения в словах

Рассмотрим фрагмент программы:

```
string x,y;
...
if (x=y)
    then
        ...
    else
        ...
```

Пусть $\mathtt{x} := \Phi(w_1, \ldots, w_n)$, $\mathtt{y} := \Psi(u_1, \ldots, u_m)$, где w_i, u_i — строковые параметры. При попытке проверки равенства $\mathtt{x}$ и $\mathtt{y}$ возникнет уравнение в словах

$$\Phi(w_1, \ldots, w_n) = \Psi(u_1, \ldots, u_m).$$

Пример: $\mathtt{x}\mathbf{A} = \mathtt{y}\mathbf{A}\mathtt{y}$

Верификация программ посредством преобразования

Дана программа $\mathcal{P}(u_1, \ldots, u_n)$, $\langle u_1, \ldots, u_n \rangle \in \mathfrak{D}$.

Пусть $\mathfrak{D}' \subset \mathfrak{D}$. Если преобразованием $\mathcal{P}(u'_1, \ldots, u'_n)$ с параметризованной входной точкой $\langle u'_1, \ldots, u'_n \rangle$, пробегающей множество $\mathfrak{D}' \subset \mathfrak{D}$, удается построить эквивалентную ей программу $\mathcal{P}'(u'_1, \ldots, u'_n)$ такую, что простые синтаксические свойства $\mathcal{P}'$ выявляют семантические особенности $\mathcal{P}(u'_1, \ldots, u'_n)$, неявные в $\mathcal{P}$, то можно говорить о верификации $\mathcal{P}$ на множестве данных $\mathfrak{D}'$ посредством преобразования.

Пример

Если программа $\mathcal{P}$ реализует предикат, $\mathcal{P}'(u'_1, \ldots, u'_n)$ может не содержать `return` **T** (либо `return` **F**) $\Rightarrow$ доказано, что $\mathcal{P}$ реализует тривиальный предикат на $\mathfrak{D}'$.

Псевдокод функционального языка над строками

Рассмотрим программу.

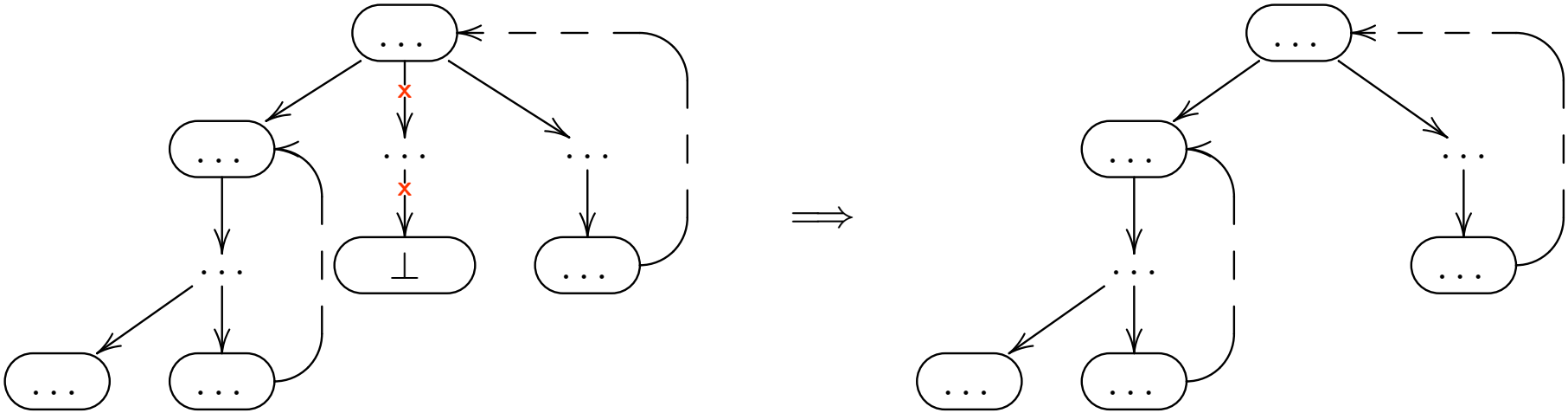
$$\boxed{\begin{array}{lcl} \texttt{Prefix}(\texttt{x}, \texttt{x}{+}{+}\texttt{y}) & = & \mathbf{T}; \\ \texttt{Prefix}(\texttt{x}, \texttt{y}) & = & \mathbf{F}; \end{array}}$$

Переменные $\texttt{x}$, $\texttt{y}$ — строкового типа. Входная точка: $\texttt{Prefix}(w_1, w_2)$.

Первое правило `Prefix` содержит повторные вхождения переменных. При попытке подстановки в его образец параметров w_1 и w_2 возникнет уравнение в словах $w_2 = w_1{+}{+}u$ (u — новый параметр).

Далее ассоциативный оператор приписывания $+{+}$ опускается.

Краткая справка о суперкомпиляции



Суперкомпиляция — это способ преобразования программ, основанный на развертке и свертке дерева параметризованных состояний программы. Предложена В.Ф. Турчиным в 1970-х годах для языка Рефал, оперирующего строками.

Частный случай верификации программы посредством суперкомпиляции — обрезка недостижимых ветвей дерева развертки ее состояний.

Суперкомпиляция выражения AllA(GenA(w))

Входная точка: AllA(GenA(w)).

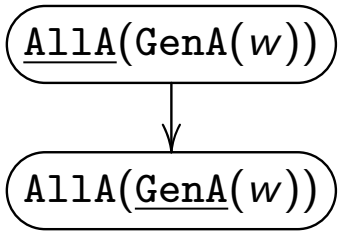
```
GenA(ε)     =  ε;
GenA(c x)   =  A GenA(x);

AllA(ε)     =  T;
AllA(A x)   =  AllA(x);
AllA(c x)   =  F;
```

Переменная c — типа символ (char). Переменная x — типа строка. Символ ε — псевдокод пустой строки, иногда опускается.

Программа реализует тривиальный предикат на данной входной точке. Доказывается суперкомпиляцией при нормальном порядке вычислений.

Развертка и перестройка стека

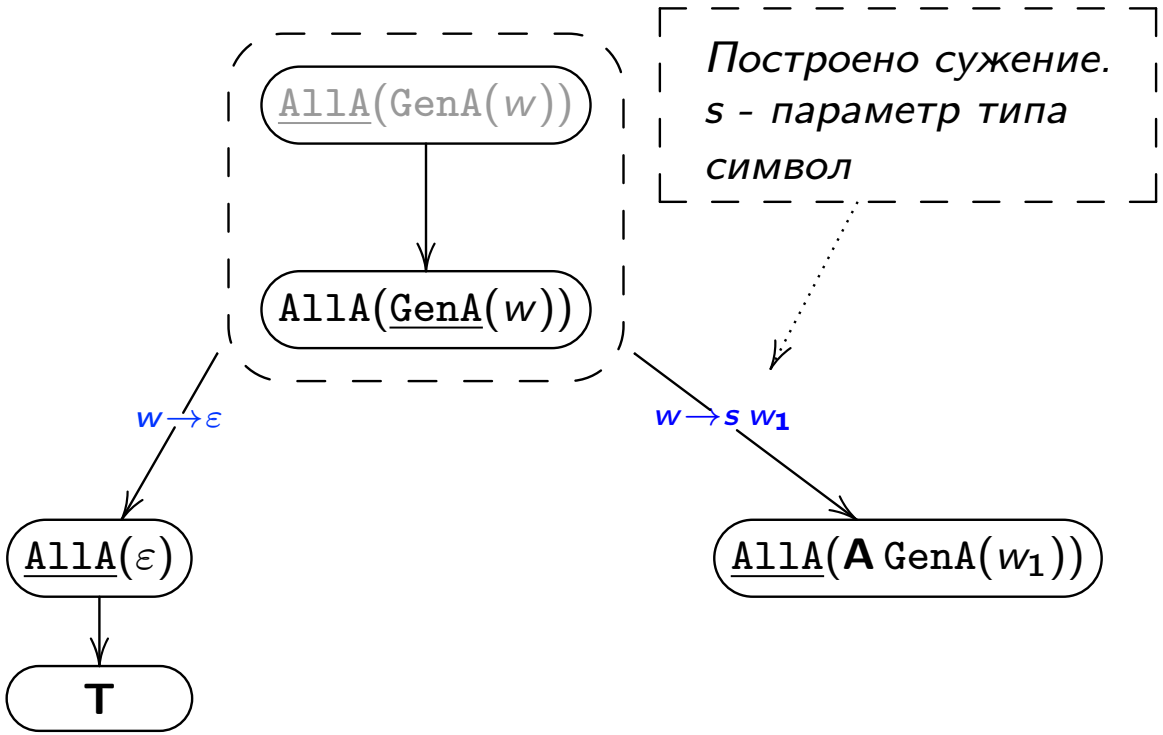


$$
\begin{array}{lcl}
\texttt{GenA}(\varepsilon) & = & \varepsilon; \\
\texttt{GenA}(\texttt{c x}) & = & \textbf{A}\,\texttt{GenA}(\texttt{x}); \\
 & & \\
\texttt{AllA}(\varepsilon) & = & \textbf{T}; \\
\texttt{AllA}(\textbf{A}\,\texttt{x}) & = & \texttt{AllA}(\texttt{x}); \\
\texttt{AllA}(\texttt{c x}) & = & \textbf{F};
\end{array}
$$

Стек перестроен.

Далее рассматривается только перестроенный стек.

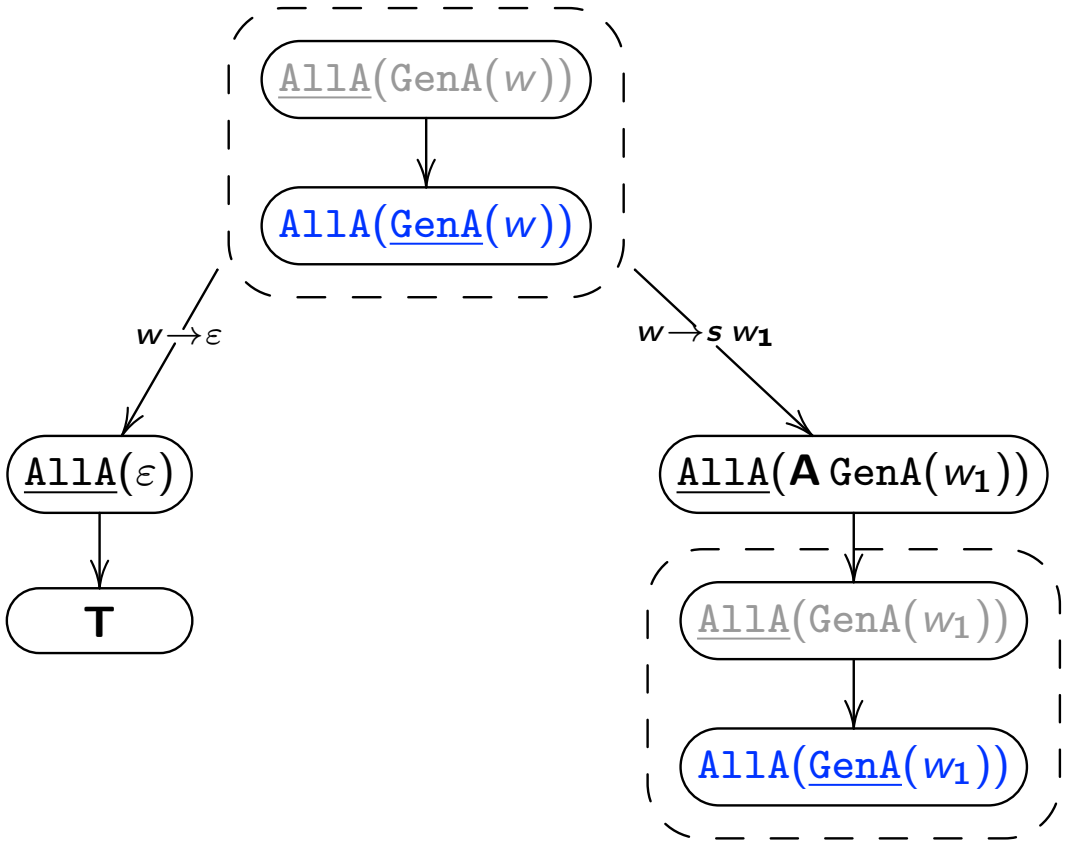
Построение сужений на параметры



$$
\begin{array}{lcl}
\texttt{GenA}(\varepsilon) & = & \varepsilon; \\
\texttt{GenA}(\texttt{c x}) & = & \textbf{A}\ \texttt{GenA}(\texttt{x}); \\
 & & \\
\texttt{AllA}(\varepsilon) & = & \textbf{T}; \\
\texttt{AllA}(\textbf{A x}) & = & \texttt{AllA}(\texttt{x}); \\
\texttt{AllA}(\texttt{c x}) & = & \textbf{F};
\end{array}
$$

Одна из ветвей вычисления состояний программы полностью развернута.

Построение вложений

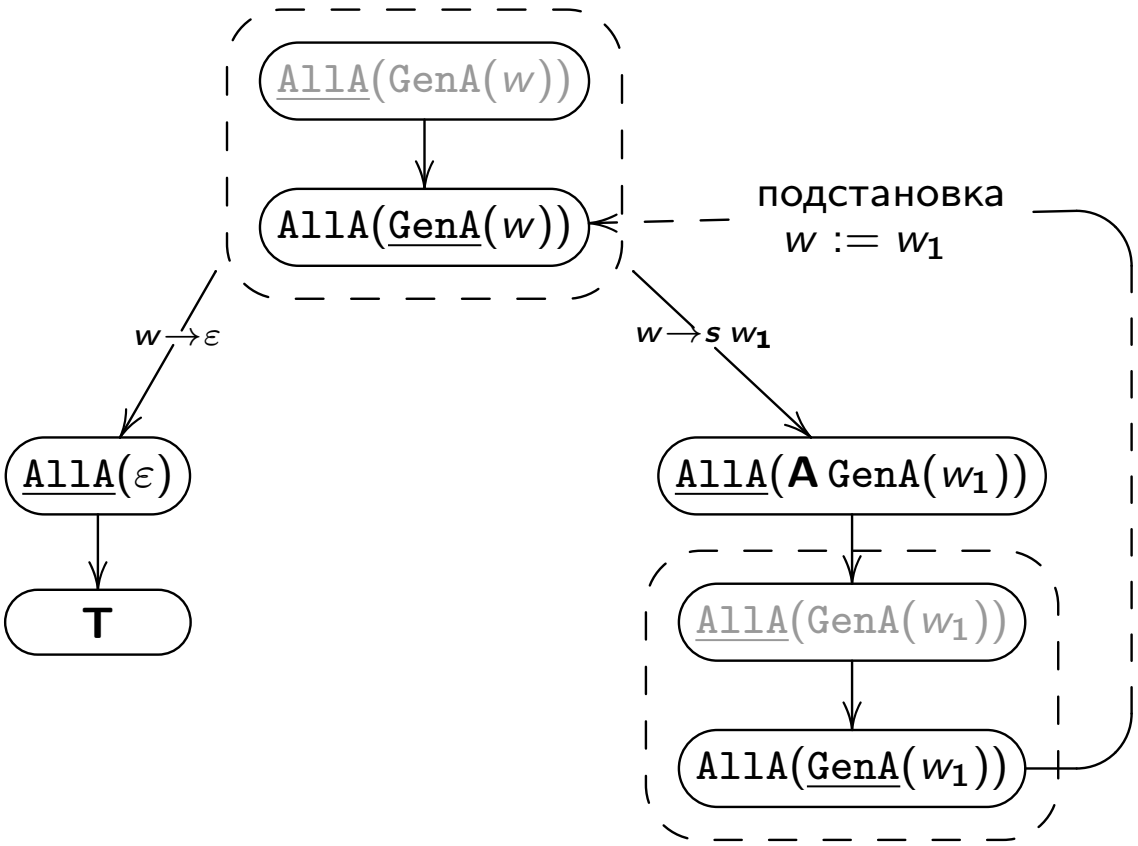



```
GenA(ε)     =  ε;
GenA(c x)   =  A GenA(x);

AllA(ε)     =  T;
AllA(A x)   =  AllA(x);
AllA(c x)   =  F;
```

Выражение AllA(GenA(w_1)) повторяет выражение AllA(GenA(w)) с точностью до переименования параметров.

Верификация посредством суперкомпиляции



$$
\begin{array}{lcl}
\texttt{GenA}(\varepsilon) & = & \varepsilon; \\
\texttt{GenA}(\texttt{c x}) & = & \textbf{A}\ \texttt{GenA}(\texttt{x}); \\
 & & \\
\texttt{AllA}(\varepsilon) & = & \textbf{T}; \\
\texttt{AllA}(\textbf{A x}) & = & \texttt{AllA}(\texttt{x}); \\
\texttt{AllA}(\texttt{c x}) & = & \textbf{F};
\end{array}
$$

Все ветви заканчиваются листьями с пассивными выражениями или зацикливаниями. Граф вычисления параметризованных состояний программы развернут.

Вершины, содержащие значение **F**, отсутствуют.

Еще один пример суперкомпиляции

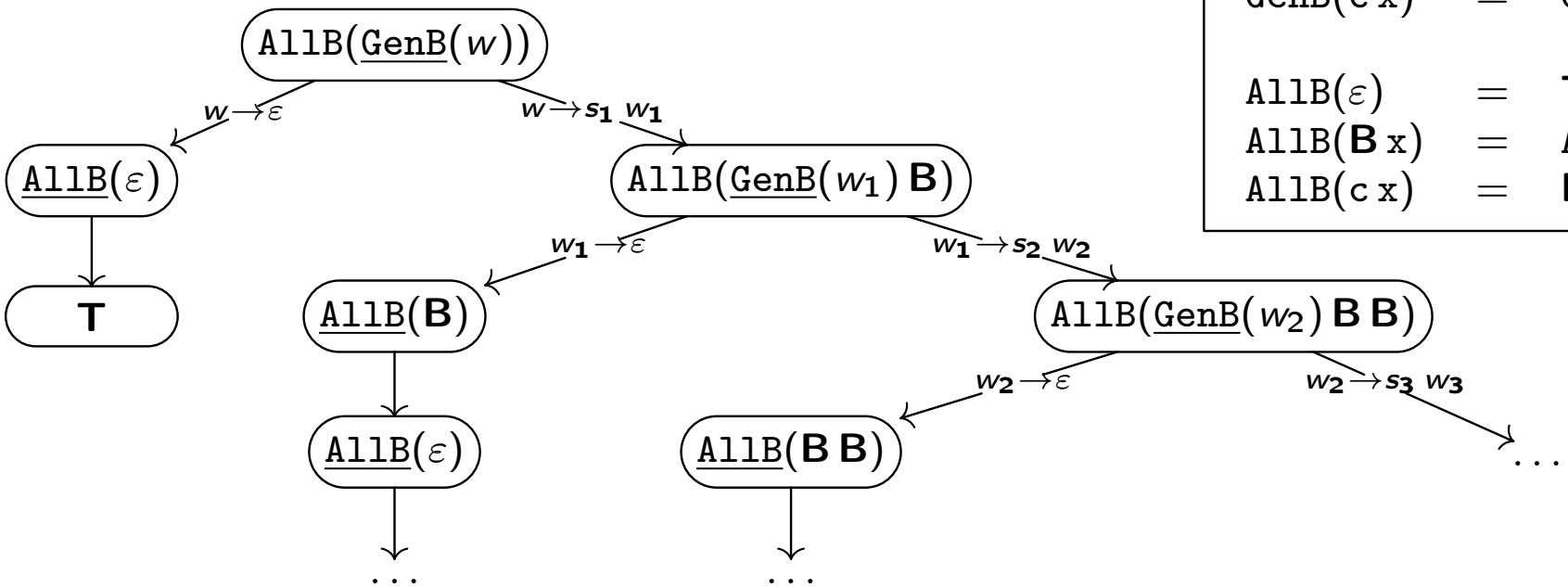
Входная точка: AllB(GenB(w)).

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

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

Нормальный порядок вычислений не помогает избавиться от нарастания букв **B** справа от вызова GenB.

Потенциально бесконечная развертка



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

Независимо от количества шагов развертки, на правой ветви дерева не удается свести выражение-потомок к выражению-предку. Дерево бесконечно.

Требуется обобщение выражений.

Обобщение

Пусть в дереве развертки выражение $\Phi_1(w_1, \ldots, w_{k_1})$ находится в узле-предке выражения $\Phi_2(w_1, \ldots, w_{k_2})$.

Обобщение выражений Φ_1, Φ_2 — выражение $\Psi(u_1, \ldots, u_n)$ и подстановки ξ_1 и ξ_2 такие, что

- $\Psi(\xi_1(u_1), \ldots, \xi_1(u_n)) \equiv \Phi_1(w_1, \ldots, w_{k_1})$,
- $\Psi(\xi_2(u_1), \ldots, \xi_2(u_n)) \equiv \Phi_2(w_1, \ldots, w_{k_2})$.

После обобщения ветвь, идущая из узла с $\Phi_1(w_1, \ldots, w_{k_1})$, удаляется, а выражение в узле заменяется на let -узел:

$$\text{let } u_1 := \xi_1(u_1), \ \ldots, \ u_n := \xi_1(u_n) \text{ in } \ \Psi(u_1, \ldots, u_n)$$

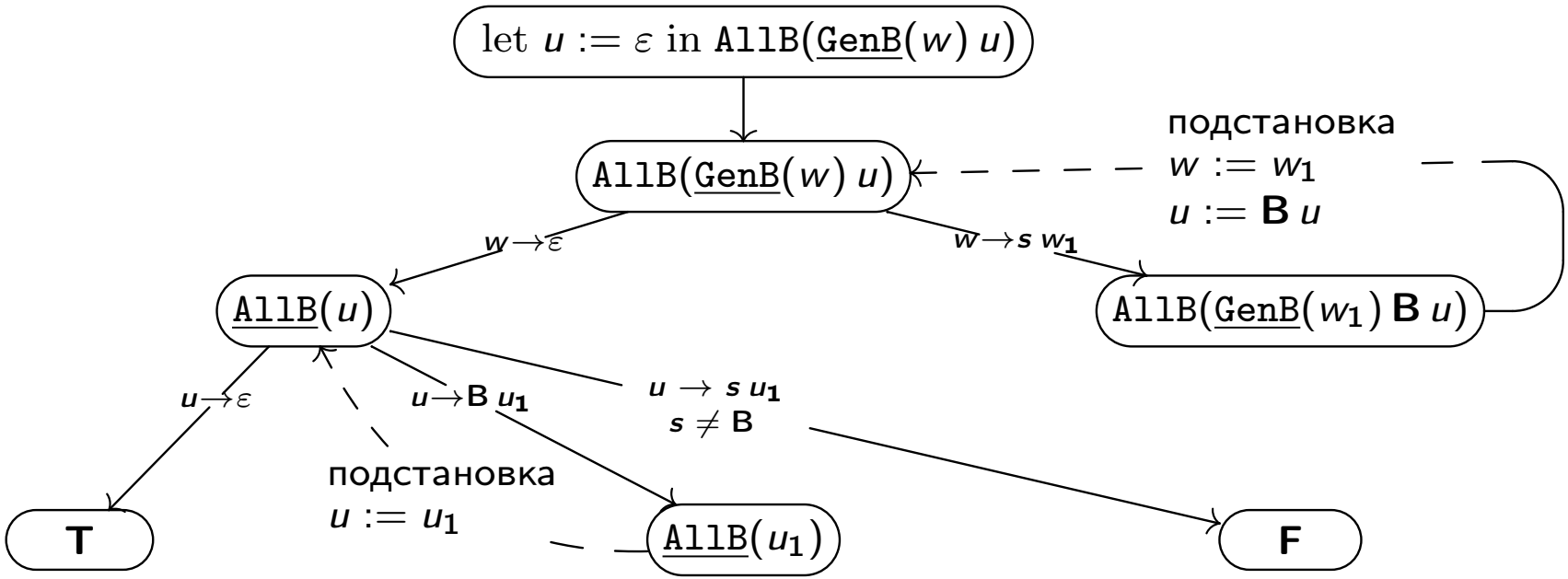
Осуществляется развертка отдельно $\Psi(u_1, \ldots, u_n)$ и (при необходимости) $\xi_1(u_i)$.

Проблема обобщения строковых данных

Не существует однозначно заданного наилучшего обобщения.

		$\Phi_1 = \mathbf{B}$	$\Phi_2 = \mathbf{B}\,\mathbf{B}\,\mathbf{B}$
	$\Psi = \mathbf{B}u$	$\xi_1(u) = \varepsilon$	$\xi_2(u) = \mathbf{B}\,\mathbf{B}$
OR	$\Psi = u\mathbf{B}$	$\xi_1(u) = \varepsilon$	$\xi_2(u) = \mathbf{B}\,\mathbf{B}$
OR	$\Psi = u\mathbf{B}u$	$\xi_1(u) = \varepsilon$	$\xi_2(u) = \mathbf{B}$
OR	$\Psi = \mathbf{B}uu$	$\xi_1(u) = \varepsilon$	$\xi_2(u) = \mathbf{B}$
OR	$\Psi = uu\mathbf{B}$	$\xi_1(u) = \varepsilon$	$\xi_2(u) = \mathbf{B}$

Построение обобщения



При обобщении выражений

$$\Phi_1 = \mathtt{AllB}(\underline{\mathtt{GenB}}(w_i)\,\mathbf{B}^n)$$
$$\Phi_2 = \mathtt{AllB}(\underline{\mathtt{GenB}}(w_{i+k})\,\mathbf{B}^{n+k})$$

возникает новый параметр u такой, что $\xi_1(u) = \varepsilon$, $\xi_2(u) = \mathbf{B}^m$.

В частности,

$$\frac{\Phi_1 = \mathtt{AllB}(\underline{\mathtt{GenB}}(w))\quad \Phi_2 = \mathtt{AllB}(\underline{\mathtt{GenB}}(w_1)\,\mathbf{B})}{\Psi = \mathtt{AllB}(\underline{\mathtt{GenB}}(w)\,u) \qquad \xi_1(u) = \varepsilon \quad \xi_2(u) = \mathbf{B}}$$

Уравнения в словах. Определения

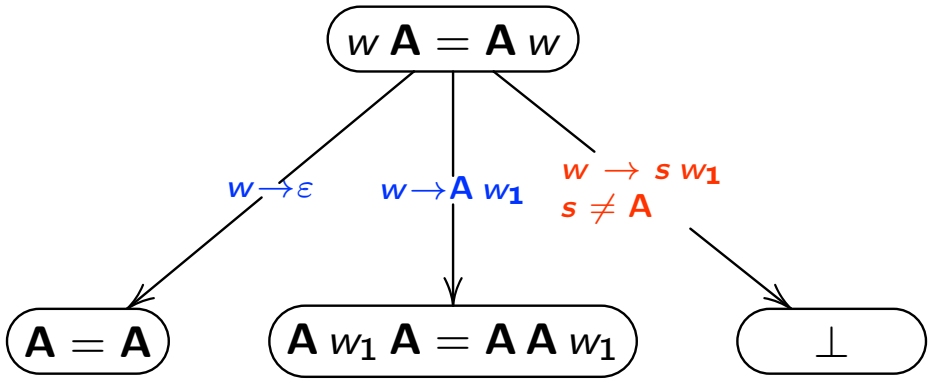
Даны алфавит констант $\mathfrak{A}$, алфавит строковых переменных $\mathfrak{V}$.
Уравнение в словах — равенство вида $\Psi = \Phi$, где $\Psi, \Phi \in \{\mathfrak{A} \cup \mathfrak{V}\}^*$.

Решить уравнение в словах — найти все такие подстановки σ переменных, входящих в $\Psi = \Phi$, что $\sigma(\Psi) \equiv \sigma(\Phi)$.

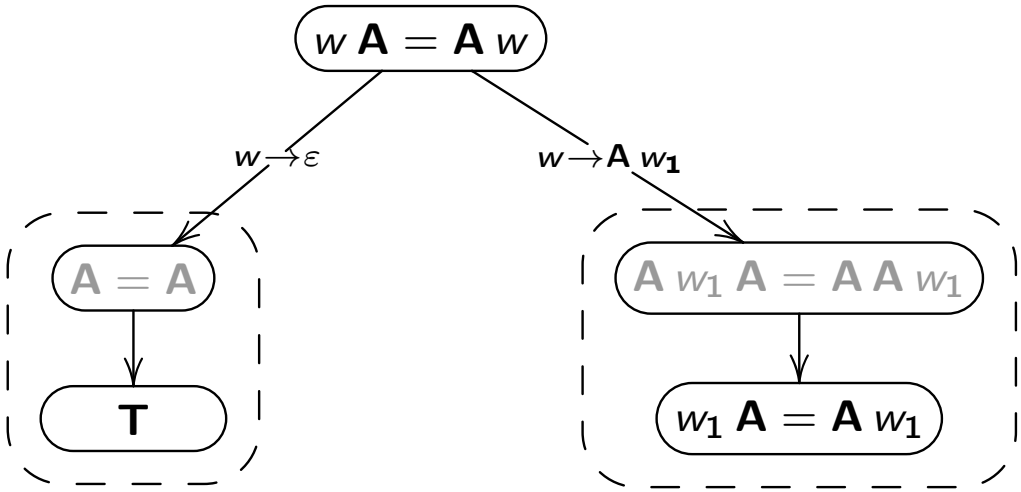
Проблема существования корней уравнений в словах разрешима. Множество решений уравнения в общем виде представляется графом сложной структуры.

Уравнение $\Psi = \Phi$ *квадратичное*, если ни одна переменная из $\mathfrak{V}$ не входит в $\Psi = \Phi$ более, чем дважды.

Дерево решения уравнения

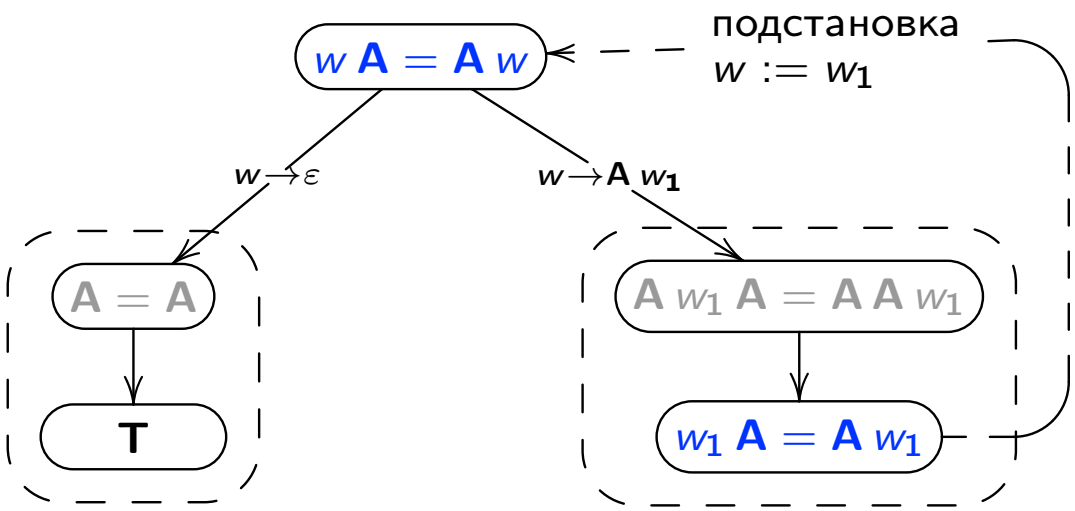


Если w не пуст, то начинается с $\mathbf{A}$, либо уравнение не выполняется.



После сужений $w \to \varepsilon$, $w \to \mathbf{A}\,w_1$ равные термы слева и справа сокращаются. Ветви дерева, приводящие к противоречию, отбрасываются.

Свертка дерева решения уравнения в граф



Уравнение $w_1\,\mathbf{A} = \mathbf{A}\,w_1$ повторяет исходное с точностью до переименования w_1 в w. Его развертка происходит точно так же.

Построен граф развертки уравнения $w\,\mathbf{A} = \mathbf{A}\,w$. Наличие в нем листьев, содержащих $\mathbf{T}$, свидетельствует о существовании корней уравнения. Множество корней: $w \in \mathbf{A}^*$.

Уравнения как язык описания свойств значений параметров

- Уравнения в словах — краткий способ записи сложных свойств значений параметров.
  - $u\,v = w\,u \Rightarrow \exists u_1, u_2, n(u = (u_1\ u_2)^n\ u_1 \& v = u_2\ u_1 \& w = u_1\ u_2)$. Здесь u_i — *строковые* параметры.
- Иногда — без очевидной альтернативы.
  - Множество решений уравнения $u\,w_1\,v = v\,w_2\,u$ нельзя описать как конечное множество формул — приписываний строк и строковых параметров, возведенных в переменные степени.
- Множество языков, описываемых уравнениями в словах, не включает и не включается в множество контекстно-свободных языков.

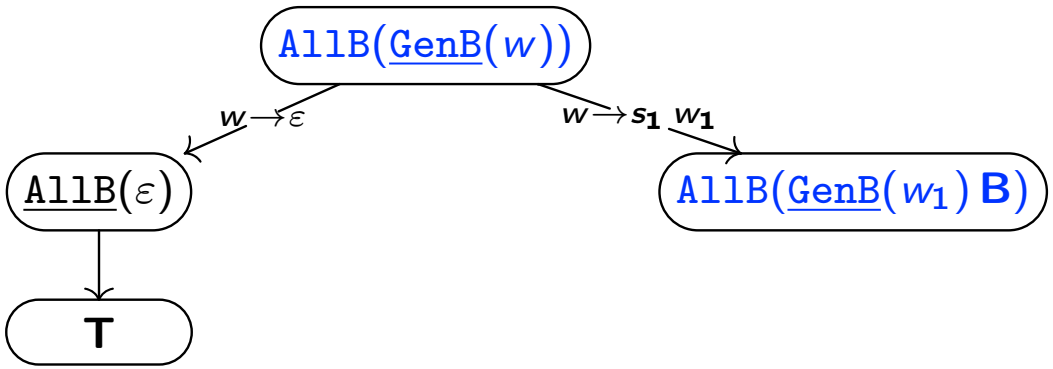
Основные особенности MSCP-A

- Основной тип данных входной программы суперкомпилятора MSCP-A— строка (с встроенной ассоциативной операцией приписывания).

- Уравнения в словах вместе с их отрицаниями составляют особый тип условий на значения параметров и существенно используются при построении графа развертки параметризованных состояний программы.

$$
\begin{array}{lll}
[\textit{Уравнение}] & ::= & [\textit{Пассивное выражение}] = [\textit{Пассивное выражение}] \\
[\textit{Формула}] & ::= & \varepsilon \mid [\textit{Уравнение}] \mid \neg[\textit{Уравнение}] \\
 & & \mid [\textit{Формула}] \ \& \ [\textit{Формула}]
\end{array}
$$

Суперкомпиляция AllB(GenB(w)) в MSCP-A



$$
\begin{array}{lcl}
\mathtt{GenB}(\varepsilon) & = & \varepsilon; \\
\mathtt{GenB}(\mathtt{c}\,\mathtt{x}) & = & \mathtt{GenB}(\mathtt{x})\,\mathbf{B}; \\
 & & \\
\mathtt{AllB}(\varepsilon) & = & \mathbf{T}; \\
\mathtt{AllB}(\mathbf{B}\,\mathtt{x}) & = & \mathtt{AllB}(\mathtt{x}); \\
\mathtt{AllB}(\mathtt{c}\,\mathtt{x}) & = & \mathbf{F};
\end{array}
$$

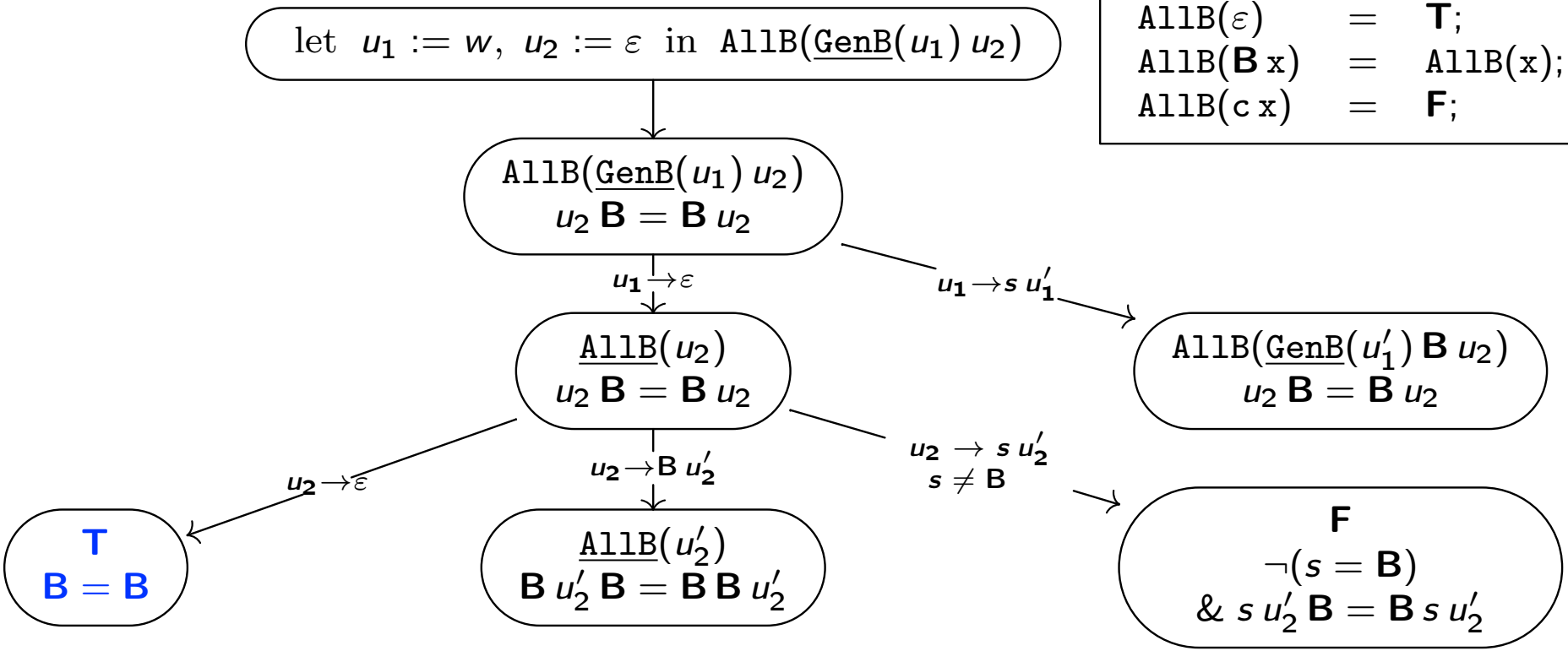
Выражения $\mathtt{AllB}(\underline{\mathtt{GenB}}(w))$ и $\mathtt{AllB}(\underline{\mathtt{GenB}}(w_1)\,\mathbf{B})$ распознаются как похожие. Строится обобщенное выражение $\mathtt{AllB}(\underline{\mathtt{GenB}}(u_1)\,u_2)$ и подстановка

$$
\begin{array}{ll}
\xi_1(u_1) = w & \xi_1(u_2) = \varepsilon \\
\xi_2(u_1) = w_1 & \xi_2(u_2) = \mathbf{B}
\end{array}
$$

В языке уравнений в словах построенное обобщение удовлетворяет условию $u_2\,\mathbf{B} = \mathbf{B}\,u_2$, поскольку $\xi_1(u_2\,\mathbf{B}) = \xi_1(\mathbf{B}\,u_2)$ и $\xi_2(u_2\,\mathbf{B}) = \xi_2(\mathbf{B}\,u_2)$.

Конфигурация: стек + формула языка уравнений

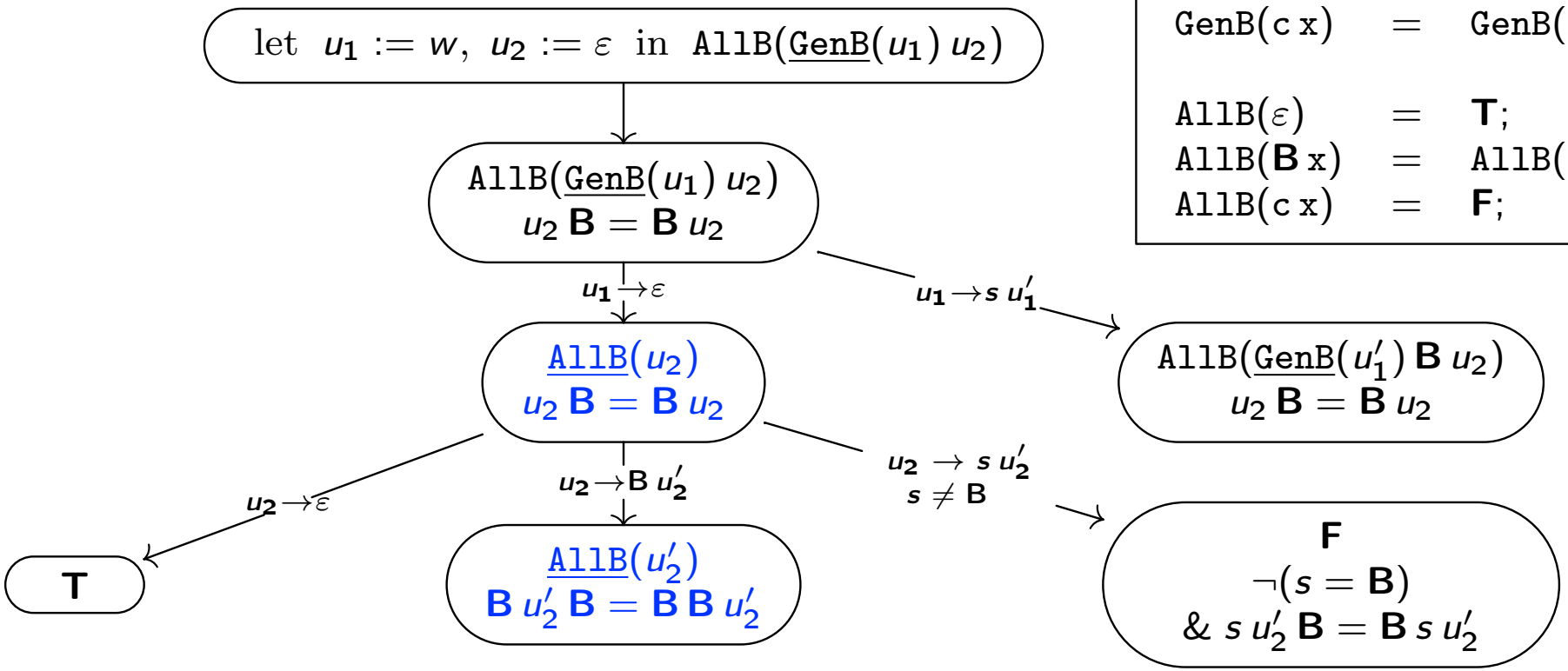
$$
\begin{array}{lcl}
\mathtt{GenB}(\varepsilon) & = & \varepsilon; \\
\mathtt{GenB}(\mathtt{c}\ \mathtt{x}) & = & \mathtt{GenB}(\mathtt{x})\ \mathbf{B}; \\
 & & \\
\mathtt{AllB}(\varepsilon) & = & \mathbf{T}; \\
\mathtt{AllB}(\mathbf{B}\ \mathtt{x}) & = & \mathtt{AllB}(\mathtt{x}); \\
\mathtt{AllB}(\mathtt{c}\ \mathtt{x}) & = & \mathbf{F};
\end{array}
$$



Уравнение $\mathbf{B} = \mathbf{B}$ тривиально и удаляется из конфигурации.

Вложение конфигураций

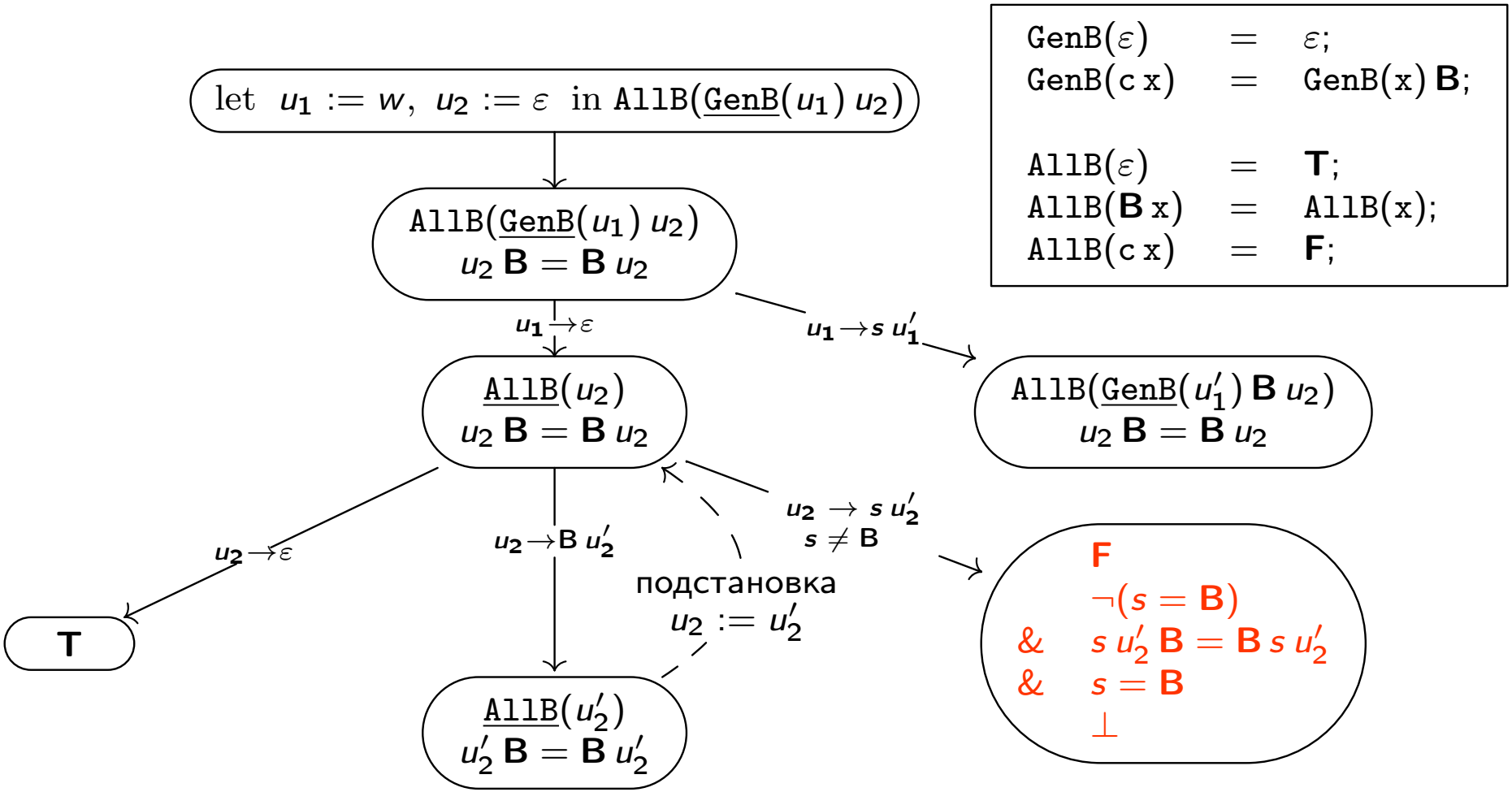
$$
\begin{array}{lcl}
\mathtt{GenB}(\varepsilon) & = & \varepsilon; \\
\mathtt{GenB}(\mathtt{c}\,\mathtt{x}) & = & \mathtt{GenB}(\mathtt{x})\,\mathbf{B}; \\
 & & \\
\mathtt{AllB}(\varepsilon) & = & \mathbf{T}; \\
\mathtt{AllB}(\mathbf{B}\,\mathtt{x}) & = & \mathtt{AllB}(\mathtt{x}); \\
\mathtt{AllB}(\mathtt{c}\,\mathtt{x}) & = & \mathbf{F};
\end{array}
$$



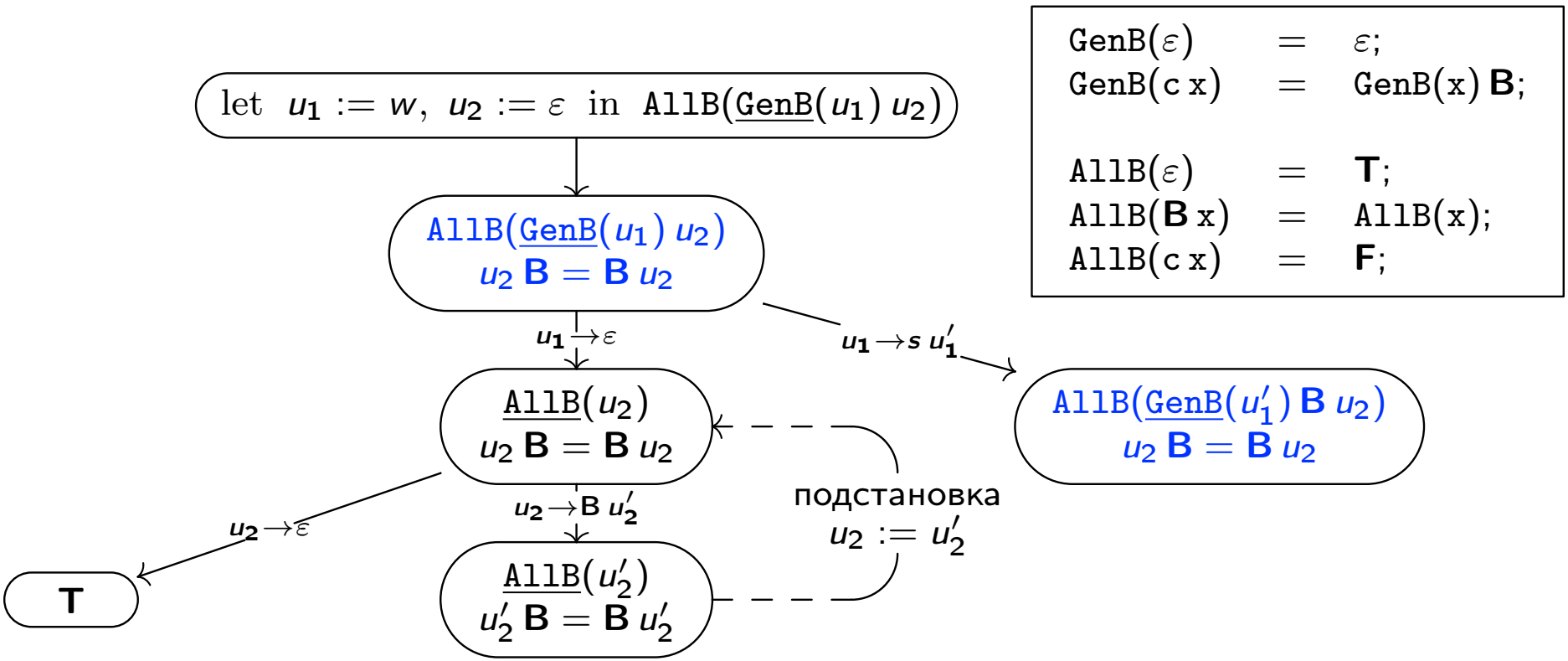
Выражение $\mathtt{AllB}(u_2)$ сводится к $\mathtt{AllB}(u_2')$ подстановкой $\xi_3(u_2) = u_2'$. Также ξ_3 сохраняет уравнение $u_2\,\mathbf{B} = \mathbf{B}\,u_2$. Может быть объявлено вложение.

$$
\begin{array}{lcl}
* \quad u_2'\,\mathbf{B} & = & \mathbf{B}\,u_2' \\
? \quad \xi_3(u_2\,\mathbf{B}) & = & \xi_3(\mathbf{B}\,u_2) \\
? \quad u_2'\,\mathbf{B} & = & \mathbf{B}\,u_2' \\
\quad \mathbf{T} & &
\end{array}
$$

Анализ системы уравнений



Конфигурация по ветви $u_2 \to s\,u_2'$ с условием $s \neq \mathbf{B}$ противоречива. Ветвь **F** отсекается.

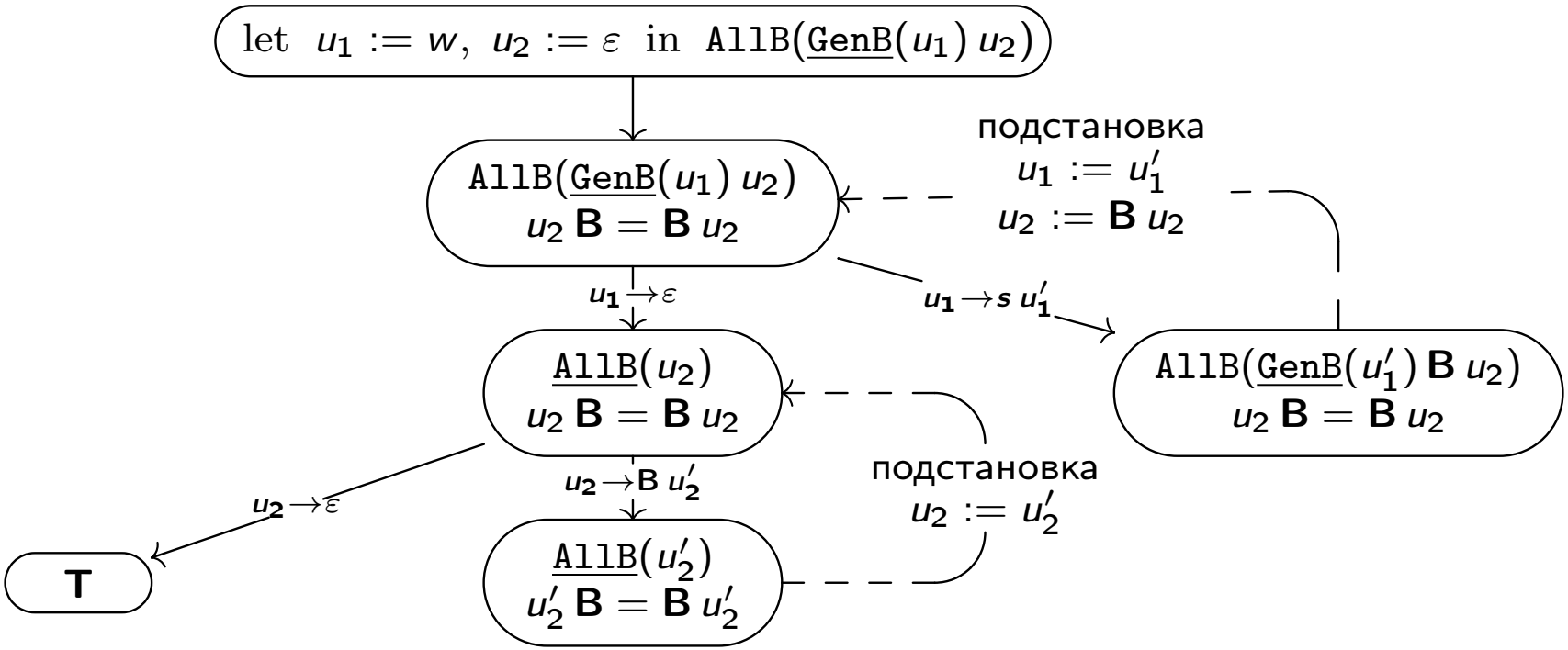


$$
\begin{array}{lcl}
\mathtt{GenB}(\varepsilon) & = & \varepsilon; \\
\mathtt{GenB}(\mathtt{c}\,\mathtt{x}) & = & \mathtt{GenB}(\mathtt{x})\,\mathbf{B}; \\
 & & \\
\mathtt{AllB}(\varepsilon) & = & \mathbf{T}; \\
\mathtt{AllB}(\mathbf{B}\,\mathtt{x}) & = & \mathtt{AllB}(\mathtt{x}); \\
\mathtt{AllB}(\mathtt{c}\,\mathtt{x}) & = & \mathbf{F};
\end{array}
$$

Выражение $\mathtt{AllB}(\mathtt{GenB}(u_1)\,\mathbf{B}\,u_2)$ получается из $\mathtt{AllB}(\mathtt{GenB}(u_1)\,u_2)$ подстановкой $\xi_4(u_1) = u_1'$, $\xi_4(u_2) = \mathbf{B}\,u_2$. ξ_4 сохраняет уравнение $u_2\,\mathbf{B} = \mathbf{B}\,u_2$. Может быть объявлено вложение.

$$
\begin{array}{lcll}
* & u_2\,\mathbf{B} & = & \mathbf{B}\,u_2 \\
? & \xi_4(u_2\,\mathbf{B}) & = & \xi_4(\mathbf{B}\,u_2) \\
? & \mathbf{B}\,u_2\,\mathbf{B} & = & \mathbf{B}\,\mathbf{B}\,u_2 \\
 & \mathbf{T} & &
\end{array}
$$

Верификация посредством суперкомпиляции



В итоговом графе состояний нет ветвей, возвращающих **F**.

Доказав гипотезу $u_2\,\mathbf{B} = \mathbf{B}\,u_2$, суперкомпилятор MSCP-A верифицировал факт, что AllB(GenB(w)) реализует тождественно истинный предикат на множестве строк.

Резюме

Исходная программа:

$$
\begin{array}{lcl}
\mathtt{Go}(w) & = & \mathtt{AllB}(\mathtt{GenB}(w)) \\
 & & \\
\mathtt{GenB}(\varepsilon) & = & \varepsilon; \\
\mathtt{GenB}(\mathtt{c}\,\mathtt{x}) & = & \mathtt{GenB}(\mathtt{x})\,\mathbf{B}; \\
 & & \\
\mathtt{AllB}(\varepsilon) & = & \mathbf{T}; \\
\mathtt{AllB}(\mathbf{B}\,\mathtt{x}) & = & \mathtt{AllB}(\mathtt{x}); \\
\mathtt{AllB}(\mathtt{c}\,\mathtt{x}) & = & \mathbf{F};
\end{array}
$$

Остаточная программа:

$$
\begin{array}{lcl}
\mathtt{Go}(w) & = & \mathtt{AllB}_1(w, \varepsilon) \\
 & & \\
\mathtt{AllB}_1(\varepsilon, \mathtt{y}) & = & \mathtt{AllB}_2(\mathtt{y}); \\
\mathtt{AllB}_1(\mathtt{c}\,\mathtt{x}, \mathtt{y}) & = & \mathtt{AllB}_1(\mathtt{x}, \mathbf{B}\,\mathtt{y}); \\
 & & \\
\mathtt{AllB}_2(\varepsilon) & = & \mathbf{T}; \\
\mathtt{AllB}_2(\mathbf{B}\,\mathtt{x}) & = & \mathtt{AllB}_2(\mathtt{x});
\end{array}
$$

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

Спасибо за внимание!

Пример нерегулярной задачи на уравнения

Исходная программа:

$$
\begin{array}{lcl}
\mathtt{Go}(u, v) & = & \mathtt{Included}(u, \mathtt{Iter}(u, v)) \\
\mathtt{Iter}(\mathtt{x}, \varepsilon) & = & \varepsilon; \\
\mathtt{Iter}(\mathtt{x}, \mathtt{c}\,\mathtt{y}) & = & \mathtt{Iter}(\mathtt{x}, \mathtt{y})\,\mathtt{x}; \\
\mathtt{Included}(\mathtt{x}, \mathtt{x}\,\mathtt{y}) & = & \mathbf{T}; \\
\mathtt{Included}(\mathtt{x}, \mathtt{c}\,\mathtt{y}) & = & \mathtt{Included}(\mathtt{x}, \mathtt{y}); \\
\mathtt{Included}(\mathtt{x}, \varepsilon) & = & \mathbf{F};
\end{array}
$$

Остаточная программа:

$$
\begin{array}{lcl}
\mathtt{Go}(u, v) & = & \mathtt{Included}_1(u, v, \varepsilon) \\
\mathtt{Included}_1(\mathtt{x}, \varepsilon, \mathtt{z}) & = & \mathtt{Included}_2(\mathtt{x}, \mathtt{z}); \\
\mathtt{Included}_1(\mathtt{x}, \mathtt{c}\,\mathtt{y}, \mathtt{z}) & = & \mathtt{Included}_1(\mathtt{x}, \mathtt{y}, \mathtt{x}\,\mathtt{z}); \\
\mathtt{Included}_2(\mathtt{x}, \mathtt{x}\,\mathtt{y}) & = & \mathbf{T}; \\
\mathtt{Included}_2(\mathtt{c}\,\mathtt{y}\,\mathtt{c}\,\mathtt{x}, \mathtt{c}\,\mathtt{y}) & = & \mathtt{Included}_2(\mathtt{c}\,\mathtt{y}\,\mathtt{c}\,\mathtt{x}, \mathtt{y}); \\
\mathtt{Included}_2(\mathtt{x}, \varepsilon) & = & \mathbf{F};
\end{array}
$$

Пример нерегулярной задачи на уравнения - 2

$$
\begin{array}{lcl}
\texttt{Go}(w) & = & \texttt{Middle}(w\,w, w); \\
 & & \\
\texttt{Middle}(\texttt{c}_1\,\texttt{x}\,\texttt{c}_2, \texttt{y}) & = & \texttt{Included}(\texttt{y}, \texttt{x}); \\
\texttt{Middle}(\texttt{c}, \texttt{y}) & = & \mathbf{F}; \\
\texttt{Middle}(\varepsilon, \texttt{y}) & = & \mathbf{F}; \\
 & & \\
\texttt{Included}(\texttt{x}, \texttt{x}\,\texttt{y}) & = & \mathbf{T}; \\
\texttt{Included}(\texttt{x}, \texttt{c}\,\texttt{y}) & = & \texttt{Included}(\texttt{x}, \texttt{y}); \\
\texttt{Included}(\texttt{x}, \varepsilon) & = & \mathbf{F};
\end{array}
$$

Остаточная программа MSCP-A указывает на следующее свойство:
$\texttt{Middle}(w\,w, w) = \mathbf{T} \Leftrightarrow \exists u_1, u_2 (w = u_1\,s\,u_2\,s\ \&\ u_1\,s\,u_2 = u_2\,s\,u_1)$.

Его анализ позволяет найти множество истинности предиката, заданного программой:
$\texttt{Middle}(w\,w, w) = \mathbf{T} \Leftrightarrow \exists u_0 (w = (u_0\,s)^+\,u_0\,s)$.

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 $v = \mathbf{B}uu$.
  - $u\mathbf{A}\mathbf{B}u = vs_1w\mathbf{A}wv \iff u\mathbf{A} = vs_1w$ и $\mathbf{B}u = \mathbf{A}wv$.
- Оценивать длины.
  - $u\,sv = v\mathbf{A}s \Rightarrow$ длина u равна 1.
  - $ws_1wus_2w = v\mathbf{A}wvu \Rightarrow$ 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;
  - ...

Обработка уравнений в MSCP-A