Сходимость преобразователей программ в метрическом пространстве деревьев

Екатерина Вербицкая

Лаборатория языков инструментов JetBrains

16 сентября 2019

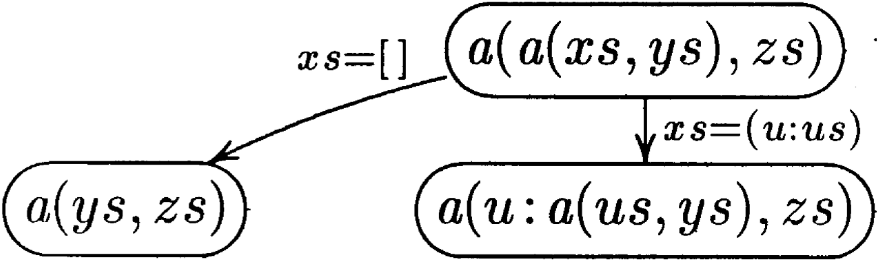
Конкатенация трех списков

$$
\begin{aligned}
a([], vs) &= vs \\
a(u:us, vs) &= u : a(us, vs)
\end{aligned}
$$

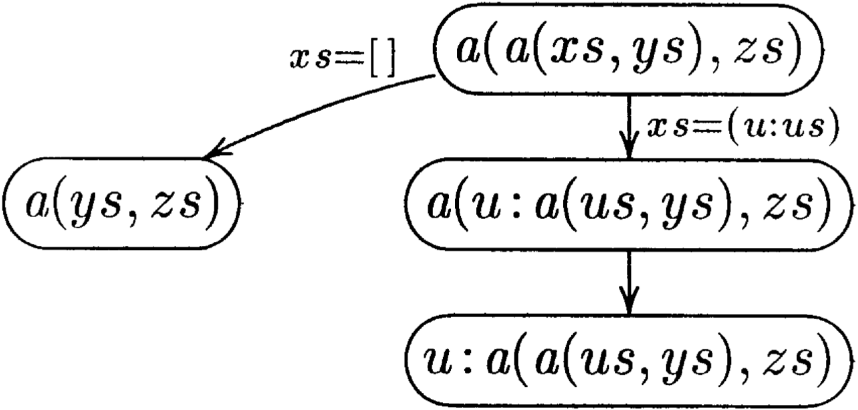
$$\boxed{a(a(xs, ys), zs)}$$

Раскрутка: переменная на месте шаблона

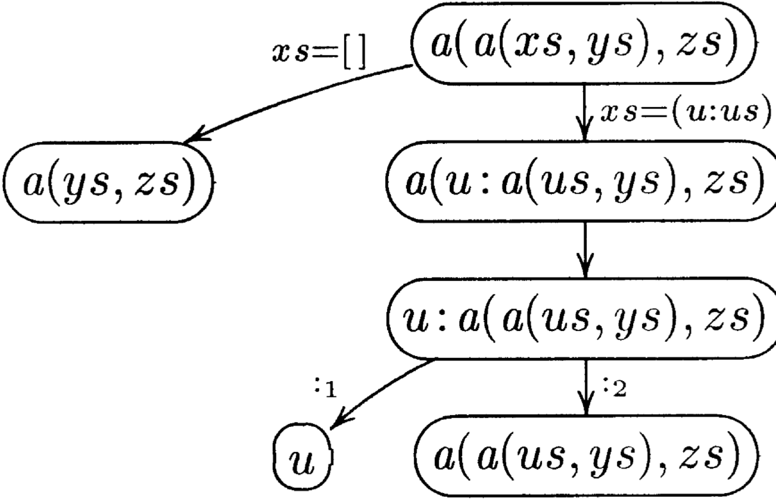
$$
\begin{aligned}
a([], vs) &= vs \\
a(u : us, vs) &= u : a(us, vs)
\end{aligned}
$$



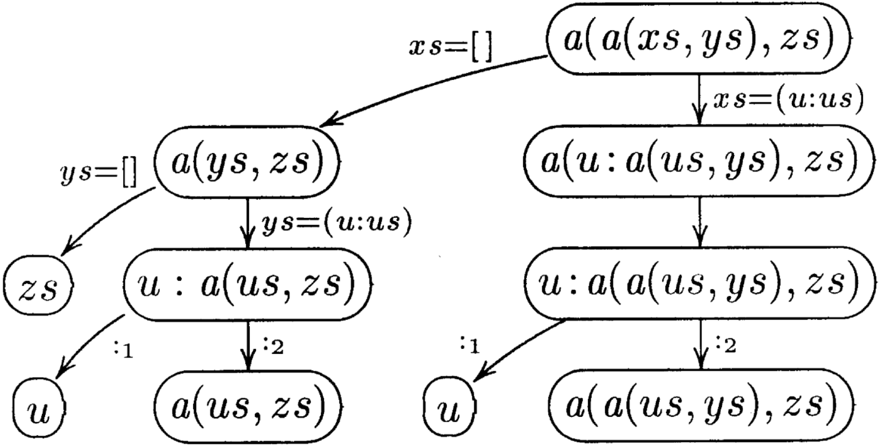
Раскрутка: шаблон инстанциирован



Раскрутка: конструктор на внешнем уровне



Расскрутка: конкатенация двух списков



Остаточная программа

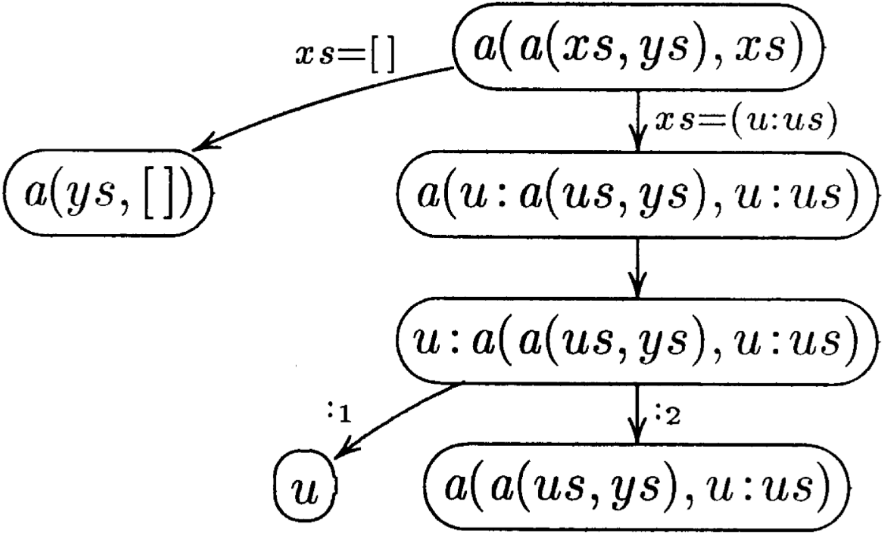
$$
\begin{aligned}
aa([], ys, zs) &= a'(ys, zs) \\
aa(u : us, ys, zs) &= u : aa(us, ys, zs)
\end{aligned}
$$

$$
\begin{aligned}
a'([], zs) &= zs \\
a'(u : us, zs) &= u : a'(us, zs)
\end{aligned}
$$

Раскрутка: одинаковые списки

$$\boxed{a(a(xs, ys), xs)}$$

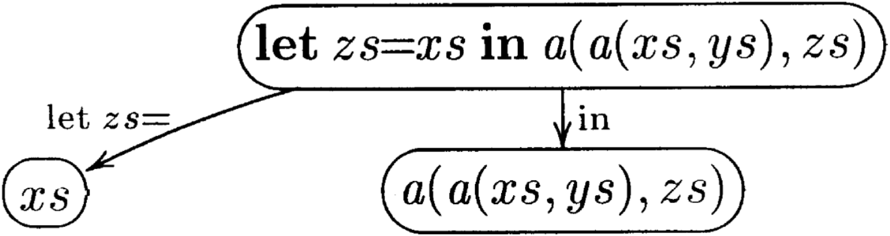
Разрастание третьего аргумента



Обобщение

$$\boxed{\mathbf{let}\ zs{=}xs\ \mathbf{in}\ a(a(xs, ys), zs)}$$

После обобщения раскрутка, как раньше



Язык

$$
\begin{array}{llll}
\mathscr{Q} \ni q ::= d_1 \ldots d_m & & & \\
\mathscr{D} \ni d ::= & f(x_1,\ldots,x_n) \triangleq e & & (\text{f-function}) \\
& \mid\ g(p_1,x_1,\ldots,x_n) \triangleq e_1 & & \\
& \qquad \vdots & & (\text{g-function}) \\
& \quad g(p_m,x_1,\ldots,x_n) \triangleq e_m & & \\
\mathscr{E} \ni e ::= x & & & (\text{variable}) \\
& \mid\ c(e_1,\ldots,e_n) & & (\text{constructor}) \\
& \mid\ f(e_1,\ldots,e_n) & & (\text{f-function call}) \\
& \mid\ g(e_0,e_1,\ldots,e_n) & & (\text{g-function call}) \\
\mathscr{P} \ni p ::= c(x_1,\ldots,x_n) & & &
\end{array}
$$

Семантика языка

$$\frac{f(x_1,\dots,x_n) \triangleq e \;\in\; q}{f(e_1,\dots,e_n) \rightarrow_{\{\}} e\{x_1 := e_1,\dots,x_n := e_n\}}$$

$$\frac{g(c(x_1,\dots,x_m),x_{m+1},\dots,x_n) \triangleq e \;\in\; q}{g(c(e_1,\dots,e_m),e_{m+1},\dots,e_n) \rightarrow_{\{\}} e\{x_1 := e_1,\dots,x_n := e_n\}}$$

$$\frac{g(p,x_1\dots,x_n) \triangleq e \;\in\; q}{g(y,e_1,\dots e_n) \rightarrow_{\{y:=p\}} e\{x_1 := e_1,\dots,x_n := e_n\}}$$

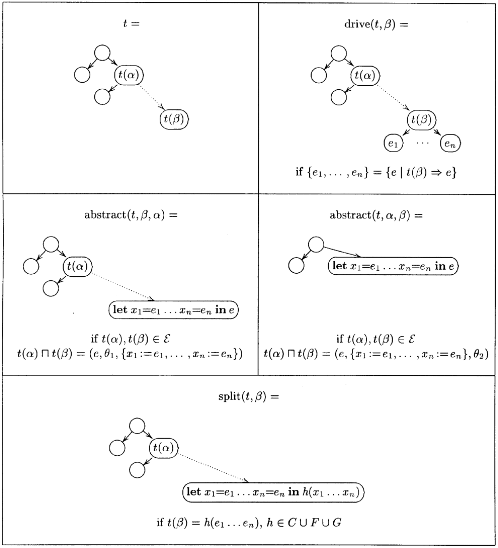
$$\frac{e \rightarrow_\theta e'}{g(e,e_1,\dots,e_n) \rightarrow_\theta g(e',e_1,\dots,e_n)}$$

$$\frac{e \rightarrow_\theta e' \quad \& \quad \theta \text{ is free for } e}{e \Rightarrow e'\theta}$$

$$\frac{i \in \{1,\dots,n\}}{c(e_1,\dots,e_n) \Rightarrow e_i}$$

$$\frac{i \in \{1,\dots,n+1\}}{\textbf{let } x_1{=}e_1,\dots,x_n{=}e_n \textbf{ in } e_{n+1} \Rightarrow e_i}$$

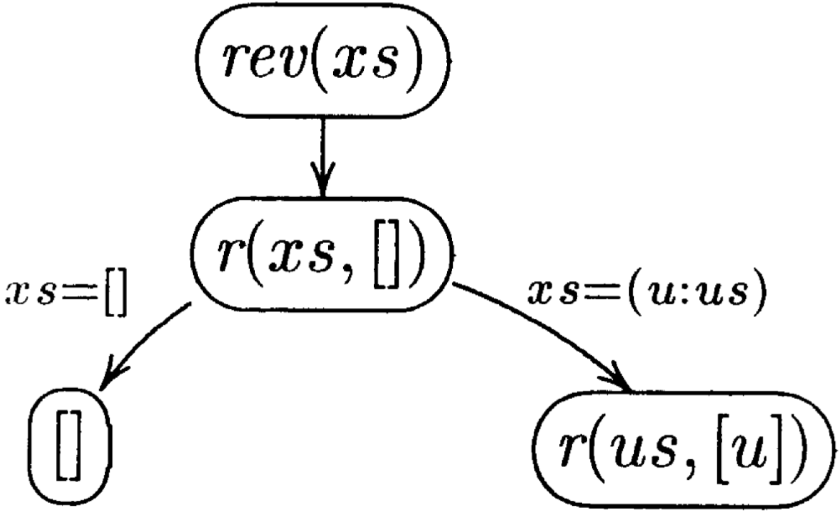
Операции суперкомпиляции

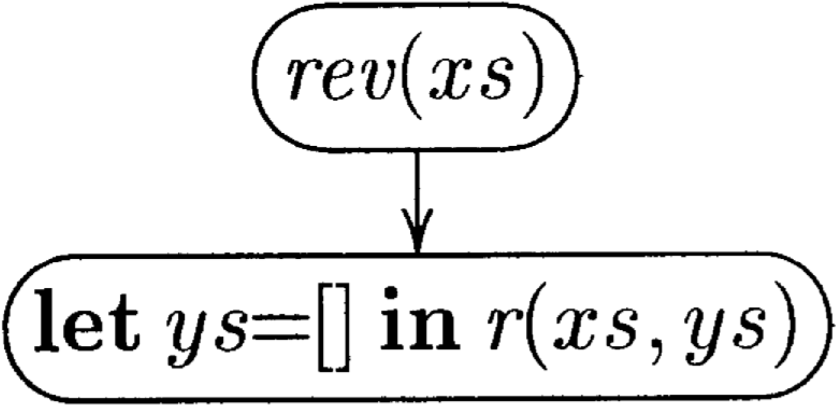


Обращение списка с аккумулирующим параметром

$$
\begin{aligned}
rev(xs) &= r(xs, []) \\
r([], vs) &= vs \\
r(u : us, vs) &= r(us, u : vs)
\end{aligned}
$$

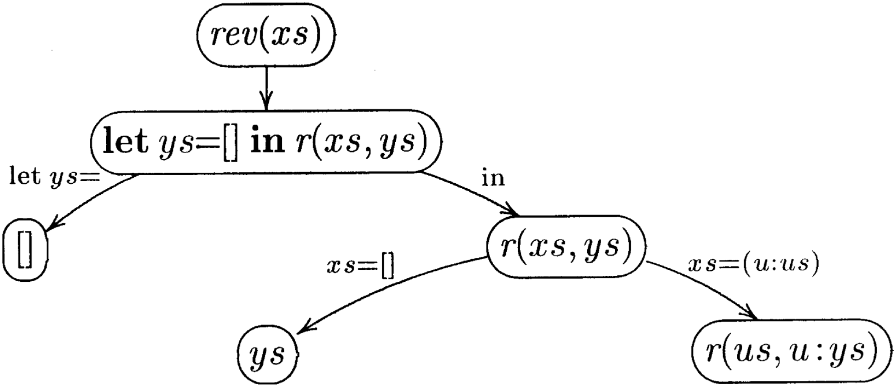
Предок вложен в лист



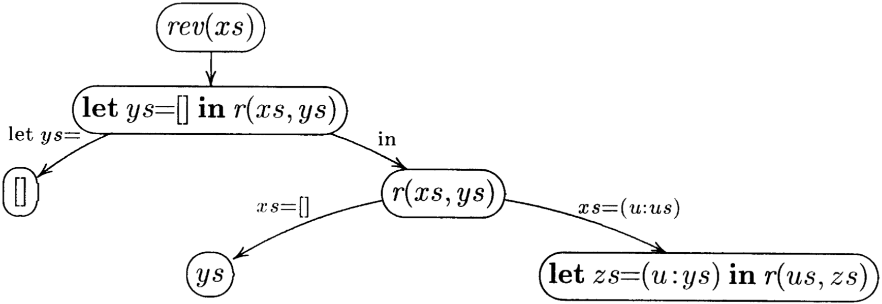


$rev(xs)$
$\mathbf{let}\ ys = []\ \mathbf{in}\ r(xs, ys)$

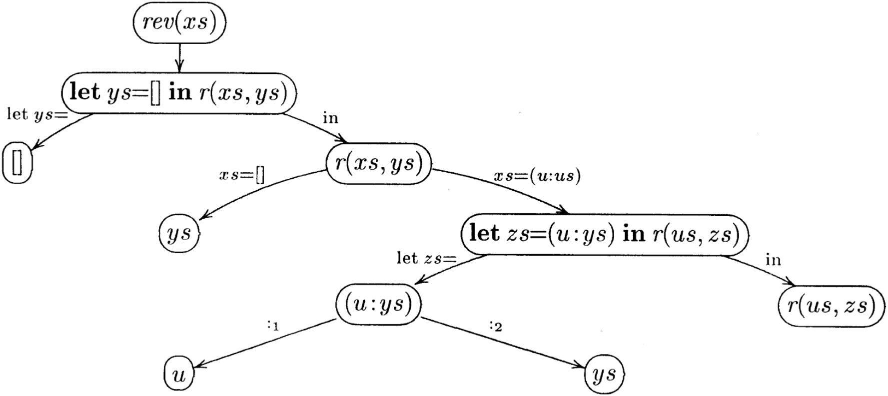
Правый нижний узел — инстанс предка



Обобщение вниз



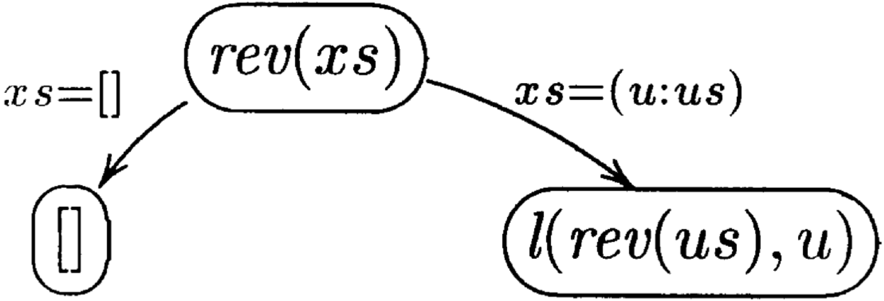
Результирующее дерево процессов



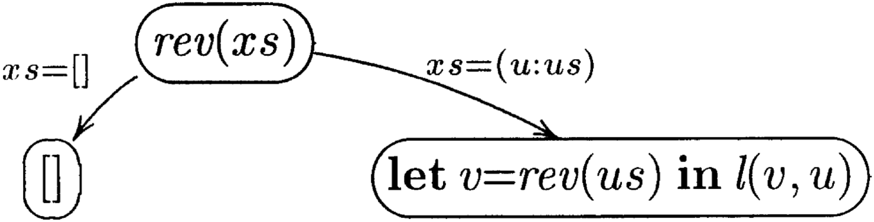
Обращение списка со вспомогательной функцией

$$
\begin{aligned}
rev([]) &= [] \\
rev(u:us) &= l(rev(us), u) \\
l([], v) &= [v] \\
l(u:us, v) &= u : l(us, v)
\end{aligned}
$$

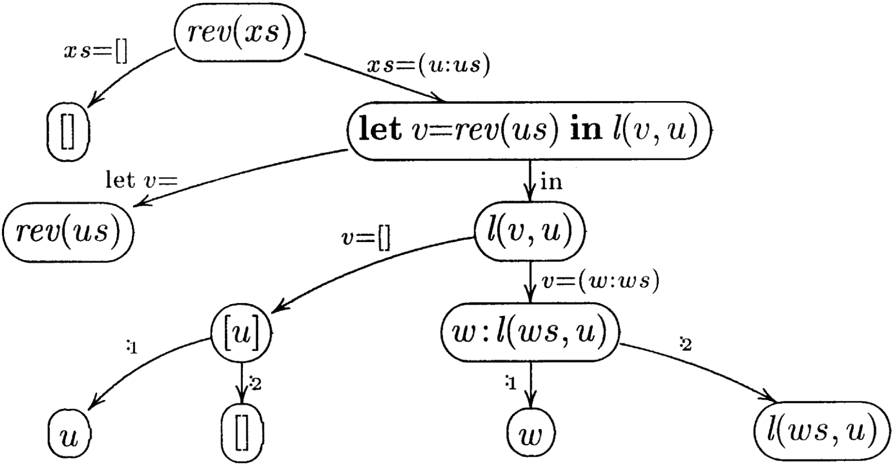
Предок вложен в лист, нет общей структуры



Обобщение: split



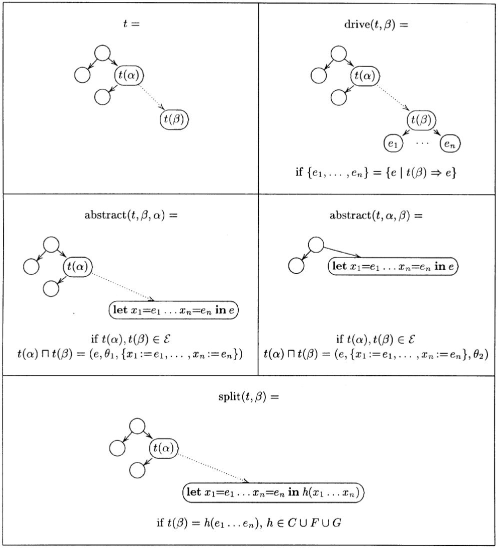
Результирующее дерево процессов



Псевдокод суперкомпилятора

$$
\begin{array}{l}
\textbf{if } \forall \alpha \in \mathrm{relanc}(t,\beta) : t(\alpha) \ntrianglelefteq t(\beta) \quad \textbf{then } P(t) = \mathrm{drive}(t,\beta) \\
\textbf{else } \textbf{begin} \\
\qquad \textbf{let } \alpha \in \mathrm{relanc}(t,\beta) \quad \text{and } t(\alpha) \trianglelefteq t(\beta). \\
\qquad \textbf{if } t(\alpha) \lessdot t(\beta) \quad \textbf{then } P(t) = \mathrm{abstract}(t,\beta,\alpha) \\
\qquad \textbf{else if } t(\alpha) \leftrightarrow t(\beta) \quad \textbf{then } P(t) = \mathrm{split}(t,\beta) \\
\qquad \textbf{else } P(t) = \mathrm{abstract}(t,\alpha,\beta). \\
\textbf{end}
\end{array}
$$

Операции суперкомпиляции



Терминируемость суперкомпилятора

Суперкомпилятор должен:

- Завершить работу на любой входной программе
  - Построить конечное дерево
- Построить замкнутое дерево: все листья имеют следующий вид:
  - Конструктор без аргументов $c()$
  - Переменная x
  - Переименование какого-то предка

Дерево

Дерево: $t : \mathbb{N}_1^* \to E$

- Не пусто
- Префиксно-замкнуто (предки всегда есть)
- Конечно ветвится
- Упорядочено (все левые братья есть)

- Множество конечных деревьев: $T(E)$
- Множество деревьев (в том числе бесконечных): $T_\infty(E)$

Абстрактный преобразователь программ

- АПП $M : T(E) \to T(E)$
- АПП M терминируется на $t \in T(E)$, если $\exists i : M^i(t) = M^{i+1}(t)$
- АПП M терминируется, если он терминируется на всех деревьях из одного узла

Метрическое пространство деревьев

- $d(t, t') = 0$, если $t = t'$
- $d(t, t') = 2^{-min\{k | t[k] \neq t'[k]\}}$, где $t[k]$ — корневое поддерево глубины не больше k

d — метрика на деревьях:

- Симметрична
- Выполняется аксиома тождества
- Выполняется неравенство треугольника

Сходящиеся последовательности

- Последовательность деревьев сходится к $t \Leftrightarrow \exists N : \forall n \geq N : t_n = t$
- Последовательность деревьев сходится к $t \Leftrightarrow \forall k : \exists N : \forall n \geq N : t_n[k] = t[k]$
- Последовательность деревьев называется последовательностью Коши $\Leftrightarrow \forall k : \exists N : \forall n \geq N : t_n[k] = t_{n+1}[k]$

- АПП M является АПП Коши, если для каждого дерева с одним узлом t последовательность $t, M(t), M^2(t), \dots$ является последовательностью Коши

Метрическое пространство деревьев является полным: любая последовательность Коши имеет предел.

Предикаты на деревьях

Предикат на деревьях: $p : T_\infty(E) \to \{1, 0\}$

- Предикат p непрерывен $\Leftrightarrow$ для любой сходящейся последовательности $t_0, t_1, \dots \in T_\infty(E)$ с бесконечным пределом t последовательность $p(t_0), p(t_1), \dots$ сходится к $p(t)$
- АПП M сохраняет (maintains) предикат p, если для любого дерева с одним узлом t и любого $i : p(M^i(t)) = 1$
- Предикат p конечен, если $p(t) = 0$ для всех бесконечных деревьев

~~Главная теорема~~ Терминируемость АПП

Пусть АПП $M : T(E) \to T(E)$ сохраняет предикат $p : T_\infty(E) \to \{1, 0\}$. Если

- M является АПП Коши и
- p конечен и непрерывен,

тогда M терминируется

Доказательство теоремы

Пусть t — дерево из одного узла, рассмотрим последовательность $t_0, t_1, \ldots$, где $t_i = M^i(t)$. По предположению она — последовательность Коши.
От противного: предполагаем, что последовательность не ограничена, тогда $|t_i| > k$. Значит, ее предел $\hat{t}$ бесконечен и $p(\hat{t}) = 0$. По непрывности $\exists N : \forall n \geq N : p(t_n) = 0$. Это противоречит сохранению предиката. Значит, последовательность ограничена:
$\exists k, I : \forall i \geq I : |t_i| \leq k$
Последовательность Коши, следовательно $\exists J : \forall j \geq J : t_j[k] = t_{j+1}[k]$.
Возьмем $N = max\{I, J\}$, тогда $\forall n \geq N$:

$$
\begin{aligned}
t_n &= t_n[k] && (|t_n| \leq k) \\
&= t_{n+1}[k] && (t_n[k] = t_{n+1}[k]) \\
&= t_{n+1} && (|t_{n+1}| \leq k)
\end{aligned}
$$

То есть последовательность сходится, а M терминируется

Послабления: слабое сохранение

АПП слабо сохраняет предикат, если $p(M^i(t)) = 0$ только для конечного числа i

В Главной теореме можно использовать слабое сохранение

Терминируемость АПП суперкомпиляции

- АПП Коши: только конечное число обобщений
- Предикат: только конечное число шагов раскрутки

АПП Коши

- $(E, \leq)$ — предпорядок, если он рефлексивен и транзитивен
- $(E, \leq)$ — well-founded, если нет бесконечных строгоупорядоченных последовательностей $e_0 > e_1 > \dots$
- $(E, \leq)$ — well-quasi-order, если в любой последовательности $e_0, e_1, \dots \exists i < j : e_i \leq e_j$

АПП является АПП Коши, если он либо добавляет потомков к листу, либо заменяет поддерево на новое, корень которого строго меньше листа.

- $\gamma \in \mathit{leaf}(t)$ & $t(\gamma) = t'(\varepsilon)$
- $t(\gamma) > t'(\varepsilon)$

Непрерывность предиката

- Предикат имеет конечный характер (finite character), если $p(t) = 1 \Leftrightarrow \forall k : p(t[k]) = 1$
- Предикат непрерывен, если он конечный и имеет конечный характер

Если p конечный и непрерывный предикат, то предикат $q(t) = p(s_1) \wedge \cdots \wedge p(s_n)$, где $\{s_1 \ldots s_n\}$ — непосредственные потомки t, является конечным и непрерывным

Важность wqo и wfqo

$(E, \leq)$ — wqo, тогда предикат p является конечным и непрерывным:

- $p(t) = 0$, если $\exists \alpha, \alpha i \beta \in dom(t) : t(\alpha) \leq t(\alpha i \beta)$
- $p(t) = 1$, иначе

$(E, \leq)$ — wfqo, тогда предикат p является конечным и непрерывным:

- $p(t) = 0$, если $\exists \alpha, \alpha i \in dom(t) : t(\alpha) \not> t(\alpha i)$
- $p(t) = 1$, иначе

Все узлы в деревьях можно разбить на классы и на каждом классе применять или wqo, или wfqo

Можно игнорировать листья

АПП в суперкомпиляции: АПП Коши

Используем теорему про добавление в листья или замену на нечто строго меньшее

Используем wfqo $e \succcurlyeq e' \Leftrightarrow e$ — выражение $\wedge e'$ — let-выражение

Раскрутка добавляет предок к листу, а любое из трех видов обобщения заменяет выражение некоторым let-выражением.

Предикат в суперкомпиляции: конечность и непрерывность

Рассматриваем предикат $q(t) = p(s_1) \wedge \dots p(s_n)$, где $\{s_1, \dots, s_n\}$ — непосредственные потомки t

- $p(t) = 0, if\ \exists \alpha, \alpha i \beta \in dom(t) : t(\alpha), t(\alpha i \beta)$ — нетривиальны[1] $\wedge t(\alpha) \trianglelefteq t(\alpha i \beta)$
- $p(t) = 0, if\ \exists \alpha, \alpha i \in dom(t) : t(\alpha), t(\alpha i)$ — тривиальны[2] $\wedge not(t(\alpha) \sqsupset t(\alpha i))$
- $p(t) = 1$ иначе

$sub(let\ x_1 = e_1 \dots x_n = e_n\ in\ e) = e\{x_1 := e_1 \dots x_n := e_n\}$

$e \sqsupseteq e' \Leftrightarrow |sub(e)| > |sub(e')| \vee (|sub(e)| = |sub(e')| \wedge sub(e) = sub(e')\theta)$

[1] не тривиальны

[2] либо конструктор, либо let-выражение