Известная проблема эффективности операционной семантики Рефала и её связь со скоростью передачи информации

Андрей П. Немытых
Институт программных систем РАН
г. Переславль-Залесский

Совместное рабочее совещание ИПС РАН и МГТУ имени Н.Э. Баумана
по функциональному языку программирования Рефал

1 июля 2024 г., Переславль-Залесский

Множество данных языка программирования Рефал

Данными языка программирования Рефал являются конечные последовательности термов.

Определение

data ::= term_data || ε
ε ::= -- пустая последовательность;
term ::= symbol || (data)
symbol ::= CHARACTER || IDENTIFIER || MACRO-DIGIT

Элементарные инструменты для конструирования данных:

Множество данных языка программирования Рефал

Данными языка программирования Рефал являются конечные последовательности термов.

Определение

data ::= term_data || ε
ε ::= -- пустая последовательность;
term ::= symbol || (data)
symbol ::= CHARACTER || IDENTIFIER || MACRO-DIGIT

Элементарные инструменты для конструирования данных:

- Приписывание одной данной последовательности термов к другой последовательности термов.

Множество данных языка программирования Рефал

Данными языка программирования Рефал являются конечные последовательности термов.

Определение

data ::= term_data || ε
ε ::= -- пустая последовательность;
term ::= symbol || (data)
symbol ::= CHARACTER || IDENTIFIER || MACRO-DIGIT

Элементарные инструменты для конструирования данных:

- Приписывание одной данной последовательности термов к другой последовательности термов.
- Построение узла (вершины) дерева.

Рефал-машина (интерпретатор)

Данными языка программирования Рефал являются конечные последовательности термов.

Элементарные инструменты для конструирования данных:

- Копирование последовательности термов.

Пример / Рабочая лента до и после преобразования

Москва пленяет пестротой "..." (Пушкин ',' 1819)

Пушкин ',' 1819 Москва пленяет пестротой "..." (Пушкин ',' 1819)

Рефал-машина (интерпретатор)

Данными языка программирования Рефал являются конечные последовательности термов.

Элементарные инструменты для конструирования данных:

- Перенос (под)последовательности термов из одного места в другое.

Пример / Рабочая лента до и после преобразования

Москва пленяет пестротой "..." (Пушкин ',' 1819)

Пушкин ',' 1819 Москва пленяет пестротой "..." ( )

Рефал-машина (интерпретатор)

Примеры:

- Приписывание одной данной последовательности термов к другой последовательности термов.

```
Concat { e.X (e.Y) = e.X e.Y; }
```

- Копирование последовательности термов.

```
Copy { e.X = e.X '#' e.X; }
```

Краткая история вопроса

Ранее все проблемы программирования в терминах языка Рефал были решены на УРА!

- Включая алгоритмически неразрешимые проблемы.
- Используя сокровенное знание.

Краткая история вопроса

Много голов было разбито об стенку, в попытках обойти законы Природы.

- Рефал++.
- Адептами Рефала++ не было предъявлено ни одного примера реального множества программ, которые бы выигрывали (в среднем) по скорости исполнения у классической простейшей реализации Рефала-5.

Краткая история вопроса

Модель вычислений: одноленточная машина Тьюринга.

- На ленте дано слово W длины n.

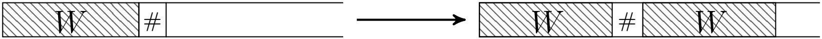

Рис.: Задача копирования.

С середины 1960-х годов известно, что решение этой задачи требует времени пропорционального n^2.

- Т.е. для любой машины Тьюринга $\mathcal{M}$, решающей эту задачу для всех слов, $\exists \epsilon, 0 < \epsilon \in \mathbb{R}$ т.ч. $\forall n \in \mathbb{N}\ \exists W$ — слово длины n, копирование которого (посредством $\mathcal{M}$) длится дольше $\epsilon \times n^2$.

Краткая история вопроса

Модель вычислений: одноленточная машина Тьюринга.

С середины 1960-х годов известно, что решение задачи требует времени пропорционального n^2, где n — длина входного слова.

- Т.е. для любой машины Тьюринга $\mathcal{M}$, решающей эту задачу для всех слов, $\exists \epsilon, 0 < \epsilon \in \mathbb{R}$, т.ч. $\forall n \in \mathbb{N}\ \exists W$ — слово длины n, копирование которого (посредством $\mathcal{M}$) длится дольше $\epsilon \times n^2$. Точная оценка:
  - $\Omega(n^2) \ni \mathrm{Time}(\mathcal{M}, W) \in \mathcal{O}(n^2), n = |W|$;
    - $\exists\, C, \epsilon > 0$, т.ч. $\epsilon n^2 < \mathrm{Time}(\mathcal{M}, W) < Cn^2, n = |W|$.

Краткая история вопроса

Модель вычислений: одноленточная машина Тьюринга.

С середины 1960-х годов известно, что решение задачи требует времени пропорционального n^2, где n — длина входного слова.

- Это один из первых результатов теории сложности вычислений.
  - Я.М. Барздинь. Сложность распознавания симметрии на машинах Тьюринга. Проблемы кибернетики. — 1965. — Т. 15. — С. 245-248.

Контрольные вопросы

Пусть фиксирована некоторая модель вычислений $\mathcal{T}$.

Дан алгоритм (или программа) $P(X)$ и значение его входного аргумента X_0.

- Чему равна сложность вычисления $P(X_0)$?
  - Т.е. можно ли указать оценку сверху сложности этого вычисления?

Контрольные вопросы

Пусть фиксирована некоторая модель вычислений $\mathcal{T}$.

Дан алгоритм (или программа) $P(X)$ и конечная последовательность значений его входного аргумента $X_1, X_2, \ldots, X_{100}$.

- Чему равна сложность вычисления $P(X)$ на этой последовательности его входных значений?
    - Т.е. можно ли указать оценку сверху сложности этого вычисления?

Контрольные вопросы

Пусть фиксирована некоторая модель вычислений $\mathcal{T}$.

Дан алгоритм (или программа) $P(X)$ и бесконечная последовательность $\{K_n \mid n \in \mathbb{N}\}$ значений его входного аргумента.

- Чему равна сложность вычисления $P(X), X \in \{K_n\}$, на этой последовательности его входных значений?
    - Что означает этот вопрос?

Верхняя оценка временно́й сложности вычислений в худшем случае

Пусть фиксирована некоторая модель вычислений $\mathcal{T}$.

Дана задача $\mathcal{F}(X)$ и бесконечная последовательность $\{K_m \mid m \in \mathbb{N}\}$ значений её входного аргумента.

- Чему равна верхняя оценка временно́й сложности вычисления $\mathcal{F}(X), X \in \{K_m\}$? Пусть $n = \mathrm{SizeOf}(X)$.
- Верхняя оценка временно́й сложности решения задачи $\mathcal{F}(X)$ в худшем случае:
  - $\mathrm{Time}_{\mathcal{T}}(P_0, X) \in \mathcal{O}(f(n))$, где $P_0(X)$ есть $\mathcal{T}$-программа, решающая задачу $\mathcal{F}(X)$, а f — функция из $\mathbb{N}$ в $\mathbb{N}$;
  - $\exists\, P_0$ — $\mathcal{T}$-программа, $\exists\, C \in \mathbb{R}, \exists\, n_0 \in \mathbb{N}$ такие, что $\forall X \in \{K_m\}$, размера $n > n_0$
    - $\mathrm{Time}_{\mathcal{T}}(P_0, X) < C \times f(n)$

Верхняя оценка временной сложности вычислений в худшем случае

Пусть $\mathcal{T}$ — модель вычислений, $X \in \{K_m\}, n = \mathrm{SizeOf}(X)$. Оценка сложности решения задачи $\mathcal{F}(X)$ в худшем случае:

- $\exists\, P_0$ — $\mathcal{T}$-программа, $\exists\, C \in \mathbb{R}, \exists\, n_0 \in \mathbb{N}$ такие, что $\forall X \in \{K_m\}$ размера $n > n_0$. $\mathrm{Time}_{\mathcal{T}}(P_0, X) < C \times f(n)$, где f — функция из $\mathbb{N}$ в $\mathbb{N}$

Таким образом:

- если существует «плохой» $X_0 \in \{K_m\}$ размера $n > n_0$, на котором
  — $C_0 f(n) < \mathrm{Time}_{\mathcal{T}}(P_0, X_0)$, то $C_0 < C$;
- Квантор всеобщности по множеству данных (множеству «первого порядка»), программа P_0 фиксирована.

Верхняя оценка временно́й сложности вычислений в худшем случае

Пусть $\mathcal{T}$ — модель вычислений, $X \in \{K_m\}, n = \mathrm{SizeOf}(X)$. Оценка сложности решения задачи $\mathcal{F}(X)$ в худшем случае:

- $\exists\, P_0$ — $\mathcal{T}$-программа, $\exists\, C \in \mathbb{R}, \exists\, n_0 \in \mathbb{N}$ такие, что $\forall X \in \{K_m\}$ размера $n > n_0$. $\mathrm{Time}_{\mathcal{T}}(P_0, X) < C \times f(n)$, где f — функция из $\mathbb{N}$ в $\mathbb{N}$

Пусть $f(n) = n^2$. Таким образом:

- если существует бесконечная подпоследовательность $X_{m_k} \in \{K_m\}$ такая, что
  — $\forall k\ (n_{m_{k+1}} = \mathrm{SizeOf}(X_{m_{k+1}})) > (n_{m_k} = \mathrm{SizeOf}(X_{m_k}))$
  $\&\ (C_0 n_{m_k}^{(2+\delta)} \leq \mathrm{Time}_{\mathcal{T}}(P_0, X_{m_k}))$, где $\delta > 0$,
  тогда верхняя оценка не может быть меньше $C_0 n^{(2+\delta)}$.

- Квантор всеобщности по множеству данных (множеству «первого порядка»), программа P_0 фиксирована.

Нижняя оценка временно́й сложности вычислений в худшем случае

Пусть фиксирована некоторая модель вычислений $\mathcal{T}$.

Дана задача $\mathcal{F}(X)$ и последовательность $\{K_m \mid m \in \mathbb{N}\}$ значений её входного аргумента.

- Чему равна нижняя оценка временно́й сложности вычисления $\mathcal{F}(X)$ на $X \in \{K_m\}$? Пусть $n = \mathrm{SizeOf}(X)$.
- Нижняя оценка временно́й сложности решения задачи $\mathcal{F}(X)$ в худшем случае:
  - $\Omega(g(n)) \ni \mathrm{Time}_{\mathcal{T}}(P, X)$ для всех $P(X)$ — $\mathcal{T}$-программ, решающих задачу $\mathcal{F}(X)$, где g — функция из $\mathbb{N}$ в $\mathbb{N}$;
  - $\forall P$ — $\mathcal{T}$-программы, решающей задачу $\mathcal{F}(X)$, $\exists\, \epsilon \in \mathbb{R}_{>0}$, $\exists\, n_0 \in \mathbb{N}$ такие, что $\forall n > n_0\ \exists\, X_0 \in \{K_m\}$, $\mathrm{SizeOf}(X_0) = n$,
    - $\epsilon \times g(n) < \mathrm{Time}_{\mathcal{T}}(P, X_0)$;

Нижняя оценка временно́й сложности вычислений в худшем случае

Пусть $\mathcal{T}$ — модель вычислений, $X \in \{K_m\}, n = \text{SizeOf}(X)$.

- Оценка сложности решения задачи $\mathcal{F}(X)$ в худшем случае:
  - $\forall P$ — $\mathcal{T}$-программы, решающей задачу $\mathcal{F}(X)$, $\exists\, \epsilon \in \mathbb{R}_{>0}$, $\exists\, n_0 \in \mathbb{N}$ такие, что $\forall n > n_0\ \exists\, X_0 \in \{K_m\}$, $\text{SizeOf}(X_0) = n$,
    - $\epsilon \times g(n) < \text{Time}_{\mathcal{T}}(P, X_0)$, где g — функция из $\mathbb{N}$ в $\mathbb{N}$;

  Таким образом:
  - если существует «хорошая» программа P_0, решающая задачу $\mathcal{F}(X)$, такая, что для всех $X \in \{K_m\}$ размера $n > n_0$
    — $C \times g(n) > \text{Time}_{\mathcal{T}}(P_0, X)$, то $C > \epsilon$;
    — Пусть $g(n) = n^2$ и $Cn^{(2+\delta)} \geq \text{Time}_{\mathcal{T}}(P_0, X)$, где $\delta > 0$, тогда нижняя оценка не может быть больше $Cn^{(2+\delta)}$.

- Квантор всеобщности по множеству программ (множеству «высшего порядка»), множеству всех алгоритмов-программ, решающих данную задачу $\mathcal{F}(X)$.

Контрольные вопросы

Пусть фиксированы некоторая модель вычислений $\mathcal{T}$ и задача $\mathcal{F}(X)$.

- Почему получение нижних оценок сложности вычисления задачи $\mathcal{F}(X)$ существенно сложнее получения верхних оценок сложности вычисления этой задачи?

Сложность вычислений определяется с точностью до «мультипликативной» константы

Пусть фиксирована некоторая модель вычислений $\mathcal{T}$.

Дана задача $\mathcal{F}(X)$ и последовательность $\{K_m \mid m \in \mathbb{N}\}$ значений её входного аргумента.

- $\Omega(g(n)) \ni \mathrm{Time}_{\mathcal{T}}(X) \in \mathcal{O}(f(n))$, где $X \in \{K_m\}$, $n = \mathrm{SizeOf}(X)$ и f, g — функции из $\mathbb{N}$ в $\mathbb{N}$;
- $\exists \epsilon, C \in \mathbb{R}_+$ такие, что $\forall X \in \{K_m\}$ размера n
  - $\epsilon \times g(n) \leq \mathrm{Time}_{\mathcal{T}}(X) \leq C \times f(n)$

Понятие «мультипликативности» зависит лишь от шкалы, которую мы привыкли использовать.

Понятие «мультипликативности» зависит лишь от шкалы, которую мы привыкли использовать

Пусть фиксирована некоторая модель вычислений $\mathcal{T}$.

Дана задача $\mathcal{F}(X)$ и последовательность $\{K_m \mid m \in \mathbb{N}\}$ значений её входного аргумента.

Переходя к логарифмической шкале, получаем определение с точностью до аддитивной константы:

- $\exists\, \epsilon, C \in \mathbb{R}_+$ такие, что $\forall X \in \{K_m\}$ размера n
  - $\epsilon \times g(n) \leq \mathrm{Time}_{\mathcal{T}}(X) \leq C \times f(n)$
  - $\log(\epsilon) + \log(g(n)) \leq \log(\mathrm{Time}_{\mathcal{T}}(X)) \leq \log(C) + \log(f(n))$

В этой шкале константные множители перед (формальным) порядком становятся значимыми (частью порядка):

- $\log(\epsilon) + 3\log(n^2) \leq \log(\mathrm{Time}_{\mathcal{T}}(X)) \leq \log(C) + 5\log(n^3)$
- $\log(\epsilon) + \log(n^6) \leq \log(\mathrm{Time}_{\mathcal{T}}(X)) \leq \log(C) + \log(n^{15})$

Сложность вычислений и скорость передачи информации

Пусть фиксирована некоторая модель вычислений $\mathcal{T}$.
Дана программа $P(X)$ и последовательность $\{K_m \mid m \in \mathbb{N}\}$ значений её входного аргумента.

Пример

$X = (m, j), P(m, j) = m \times j$, где $m, j \in \mathbb{N}$ и заданы в десятичной системе счисления.

- $(m_0, j_0) = X_0 \in \mathbb{N} \times \mathbb{N}$ содержит некоторую информацию,
  - часть которой может быть несущественной для вычисления результата $P(X_0)$;
    - Мы интересуемся значением $P(m_0, 10^{i_0}) = ?$.
      — информация о значении цифр числа m_0 несущественна;
  - без другой части результат вычисления $P(X_0)$ построить невозможно.
    — значимая информация: $\mathrm{length}(m_0)$ и значение i_0.

Сложность вычислений и скорость передачи информации

Важное замечание.

- Чем медленнее алгоритм (программа) передает (от «оператора к оператору») информацию, значимую для вычисления результата, тем медленнее этот алгоритм вычисляет результат.
- Чем больше скорость передачи информации, тем больше у алгоритма шансов быстрее вычислить результат.

Что такое информация?

Наивно интересуемся информацией, заключенной в некотором описании конечного объекта.

- В какой из двух последовательностей слов содержится больше информации:
    - В какой из двух последовательностей слов содержится больше информации:
      ПОЛЕ ПОЛЕ ПОЛЕ ПОЛЕ ПОЛЕ ПОЛЕ ПОЛЕ ПОЛЕ
      ПЛЛЕ ЛПЛП ОЛЕП ЛЛПЕ
      ?
- Сколько информации содержится в натуральном числе $n \in \mathbb{N}$?
- Много ли информации содержит десятичное представление иррационального (и даже трансцендентного) числа $\pi = 3,1415\ldots.$?

Список использованной литературы

- Н. К. Верещагин, В. А. Успенский, А. Х. Шень. Колмогоровская сложность и алгоритмическая случайность. –– МЦНМО, 2013.
- A. Shen. Kolmogorov complexity as a language // International Computer Science Symposium in Russia. 2011. – pp: 105-119.

Что такое информация?

Наивно интересуемся информацией, заключенной в некотором описании конечного объекта.

- Информация это фундаментальное свойство строки (длинного слова) W, которое практически не меняется, т.е. слабо меняется, при простых преобразованиях W.
  - Например, число вхождений подстроки '0101' в W не может быть мерой информации, поскольку, в общем случае оно может быть сильно изменено простым преобразованием $\mathrm{rev}(W)$ — реверсирования этой строки. (Такой шифр можно быстро расшифровать.)
  - Простые преобразования не способны менять фундаментальные свойства строки.

Что такое информация?

Наивно интересуемся информацией, заключенной в некотором описании конечного объекта.

- Информация это фундаментальное свойство строки (длинного слова) W, которое практически не меняется, т.е. слабо меняется, при простых преобразованиях W.
- Следовательно, наивное понятие меры информации, в общем случае, зависит от модели вычислений, как и сложность вычислений.
  - Интересные вопросы:
    — Насколько существенно она зависит от этой модели?
    — Можно ли как-то построить теорию информации, разумно абстрагируясь от модели вычислений?
    — Может ли какое-нибудь преобразование строки W существенно увеличить информацию, содержащуюся в W?

Что такое информация?

Наивно интересуемся информацией, заключенной в некотором описании конечного объекта.

- В какой из двух последовательностей слов содержится больше информации:
  - В какой из двух последовательностей слов содержится больше информации:
    ПОЛЕ ПОЛЕ ПОЛЕ ПОЛЕ ПОЛЕ ПОЛЕ ПОЛЕ ПОЛЕ
    — ПОЛЕ 8 раз
    ПЛЛЕ ЛПЛП ОЛЕП ЛЛПЕ
    — Как более кратко описать информацию в этой строке?
- Сколько информации содержится в натуральном числе $n \in \mathbb{N}$?
  - $[\log_2(n)] \leq \inf(n) \leq [\log_2(n)] + 1$, если n дано в двоичной системе счисления;
  - $[\log_{10}(n)] \leq \inf(n) \leq [\log_{10}(n)] + 1$, если n дано в десятичной системе счисления.

Что такое информация?

Наивно интересуемся информацией, заключенной в некотором описании конечного объекта.

- Информация это фундаментальное свойство строки (длинного слова) W, которое практически не меняется, т.е. слабо меняется, при простых преобразованиях W.
    - Следовательно, чем менее слово W структурировано, тем больше оно содержит информации. (Такой шифр трудно расшифровать.)
    - Больше всего информации содержится в случайно сгенерированном слове.

Информация и упаковщики/распаковщики

- Цель упаковщика (архиватора-шифровальщика): выявить во входной строке W (файле) всю существенную информацию, по которой можно восстановить W, и попытаться удалить как можно больше ту часть W, которая не является необходимой для восстановления W; построить всю существенную информацию, т.е. зашифровать W.
- Цель распаковщика (разархиватора-дешифровщика): восстановить/дешифровать исходный файл.

Информация и упаковщики/распаковщики

Пусть дано слово W. Чем меньше длина архива, построенного из W, тем лучше для данного слова W использованный архиватор. В общем случае, понятие «лучшего архиватора» не является равномерным: почти для каждого слова — свой лучший архиватор.

Теорема

Идеальной, т.е. лучшей сразу для всех входных слов (файлов), программы-архиватора не существует.

Модель вычислений: однолентонная машина Тьюринга

Пусть рабочий алфавит машина Тьюринга — бинарный (битовый) алфавит.

Какая максимальная скорость переноса информации конкретной машиной Тьюринга МТ по ленте за один такт (шаг) работы?

Мгновенная скорость переноса информации по рабочей ленте за один сдвиг головки

зависит от состояния S, в котором находится МТ и равна информации, содержащейся в S.

Модель вычислений: одноленточная машина Тьюринга

Пусть рабочий алфавит машина Тьюринга — бинарный (битовый) алфавит.

Мгновенная скорость переноса информации по рабочей ленте за один сдвиг головки

зависит от состояния S, в котором находится МТ и равна информации, содержащейся в S.

Следовательно, скорость переноса информации конкретной машиной Тьюринга МТ с m состояниями по ленте за один шаг работы не больше чем $[\log_2(m)] + 1$ бит.

Модель вычислений: одноленточная машина Тьюринга

- На ленте дано слово W длины n.

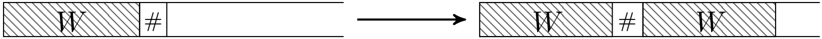

Рис.: Задача копирования $\mathcal{Q}$.

Теорема

$\mathrm{Time}_{\mathcal{T}}(\mathcal{Q}, W) \in \Omega(n^2)$.

Модель вычислений: одноленточная машина Тьюринга

Теорема

$\mathrm{Time}_{\mathcal{T}}(\mathcal{Q}, W) \in \Omega(n^2)$.

Доказательство (Версия A):

- Рассмотрим частный случай – слово $|W| = n/2$:

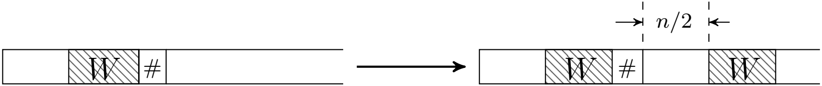

Рис.: Частный случай: первая половина слова W является пустой.

Доказательство (Версия А)

- Рассмотрим частный случай – слово $|W| = n/2$:

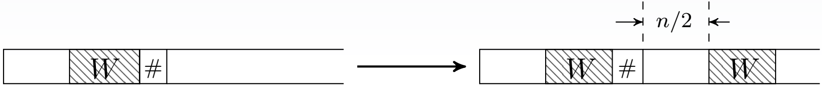

Рис.: Частный случай: первая половина слова W является пустой.

Чтобы скопировать слово W, машина МТ должна перенести $n/2$ бит информации правее дырки длины $n/2$. Для этого требуется не менее $\epsilon \times (n/2)^2$ шагов МТ, где константа ϵ не меньше скорости переноса информации за один шаг машины МТ.

Следовательно, $\mathrm{Time}_{\mathcal{T}}(\mathcal{Q}, W) \in \Omega(n^2)$. $\square$

Доказательство (Версия B)

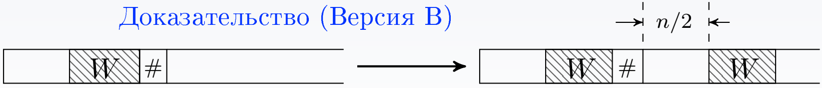

Рис.: Частный случай: первая половина слова W является пустой.

Установим таможенный пост внутри дырки.

Таможенники записывают состояния машины, в которых она пересекает границу слева направо. Содержание журнала записей достаточно для восстановления слова W за территорией государства; так как поведение ТМ за границей полностью определяется содержанием журнала.

Следовательно, длина записи в журнале должна быть $\Omega(n)$.

Это рассуждение верно для каждой клетки в дырке, где можно установить таможенный пост. И сумма длин записей по всем таким таможням есть нижняя оценка для числа шагов машины МТ.

Следовательно, $\mathrm{Time}_{\mathcal{T}}(\mathcal{Q}, W) \in \Omega(n^2)$.

Рефал-машина vs. машина Тьюринга

- В контексте задачи копирования модель вычислений машины Тьюринга почти совпадает с моделью вычислений Рефал-машины, операционной семантикой интерпретатора.
- В случае Рефал-машины:
  - элементарные ячейки поля памяти, представляющие термы, соответствуют клеткам машины Тьюринга.
  - по ленте движутся две головки, которые могут передавать друг другу информацию о собственных состояниях.