

XVIII конференция “Свободное программное обеспечение в высшей школе”,  
27-29 января 2023 г., Переславль-Залесский

# Рефал: опыт формализации в преподавании информатики

Исакевич Даниил Валерьевич,  
Владимирский строительный колледж,  
ООО “БизнесСофтСервис”,  
ООО “Собственный вектор”

## Краткая характеристика ситуации

Информатика (темы: “количество информации”, “системы счисления”, “математическая логика”, “алгоритмы”, “работа с офисными приложениями”)

Студенты: первый курс колледжа, строительные специальности, увлечения – разные, большинство со слабой математической подготовкой

**Сентябрь 2021 года:** Недостаток методических наработок, опыта преподавания

**Январь 2023 года:**

Наличие и использование наработанных примеров и ответов для отдельных тем: вычисление таблиц истинности, оценка количества информации (биты, байты; кило-, мега-, гига-, тера-; триты), перевод чисел между позиционными системами счисления

Формулируются задачи, требующие знаний из разных тем

Формулируются задачи, показывающие практическую применимость тем

**Что и как было сделано за этот год?**

# Использованные инструменты

Операционная система: ALTLinux p9 Starterkit (Hypericum)

Вычислитель программ в структурном стиле: GNU Octave (аналог MATLAB)

Средство вывода в PDF: TeXLive

Рефал:

Refal-5 Version PZ (<http://www.botik.ru/pub/local/scp/refal5/>)

Рефал-M (<https://github.com/STrusov/refal-machine>)

## **Некоторые существенные особенности Рефал-M:**

Можно писать программы, читаемые как связный текст с русскими словами и словосочетаниями.

Можно применять сокращения для типов переменных:

. что-нибудь      вместо      e.Something

?цифра            вместо      s.Digit

! терм             вместо      t.Term

Ввод-вывод выражений без мешающих преобразований (сериализация массивов).

## Что сделано? Что замечено? Что дальше?

1. Автоматическая массовая генерация примеров и ответов.
2. Отказ от стандартной реализации арифметики.
3. Программа на Рефал-М – русскоязычный текст.
- 4, 5. Выводы, перспективы и вопросы.

# 1. Автоматическая массовая генерация примеров и ответов

- Вынуждает учиться
- Обеспечивает разнообразие

“В самом деле, одна из наиболее убедительных причин использования вычислительных машин – возможность обнаруживать слабые места в наших рассуждениях. В этом отношении вычислительные машины – безжалостные критики.” (Дж. Вейценбаум, “Возможности вычислительных машин и человеческий разум”)

Выражение номер 5.114

$$\overline{C} \wedge ((B \equiv B) \rightarrow (A \vee C))$$

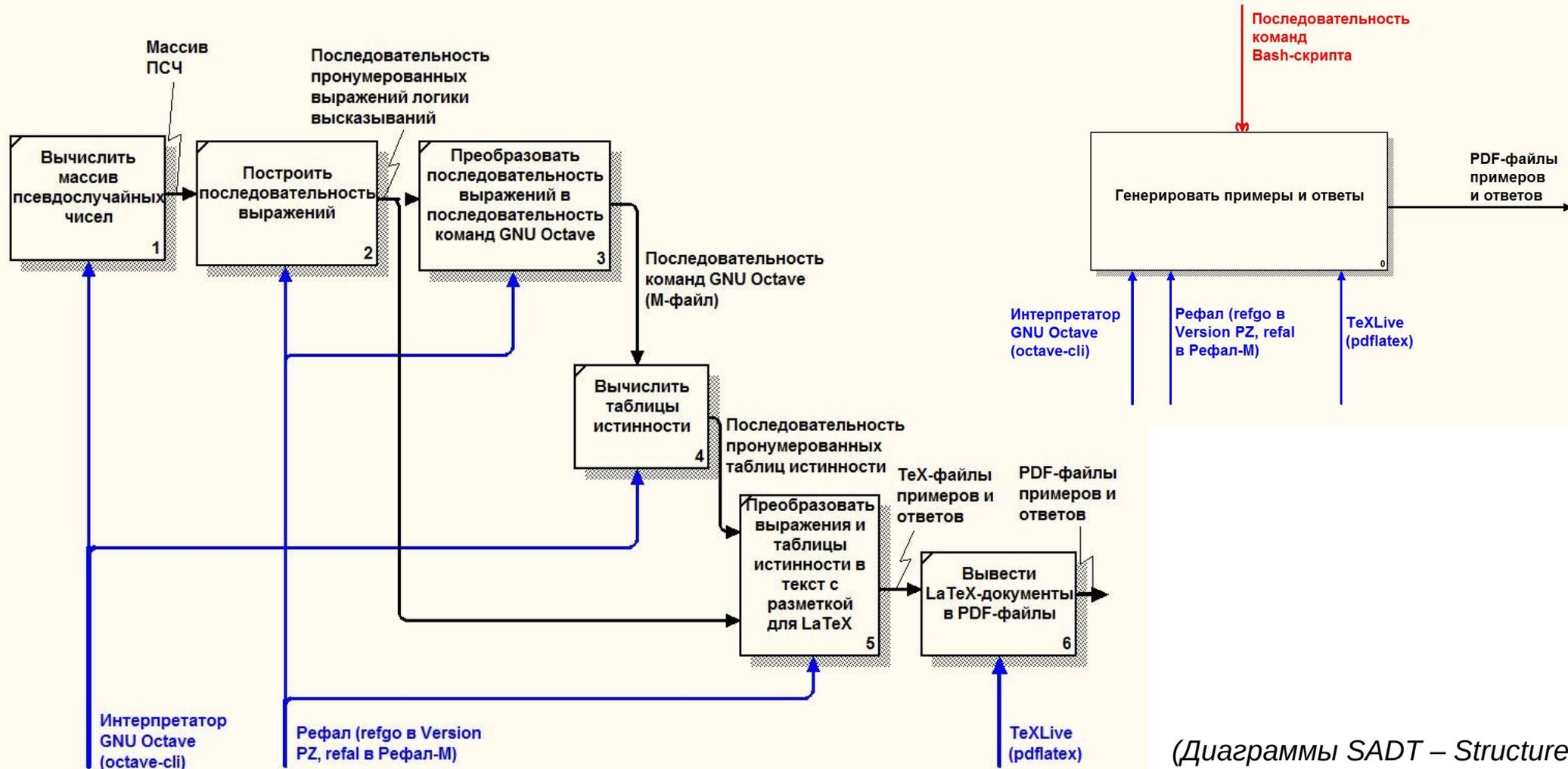
Выражение номер 5.72

$$\overline{\overline{B \vee B \vee B \vee C \vee A}}$$

Выражение номер 5.151

$$C \rightarrow ((C \vee A \vee (A \equiv B)) \vee A \vee (A \wedge A \wedge B \wedge A))$$

# 1. Генерация выражений и таблиц истинности: процесс



(Диаграммы SADT – Structured Analysis and Design Technique)

# 1.Генерация выражения на Refal-5 Version PZ

```
Tree {
  0 = ' ' <Nth <Mod <Dg 1> <TreeNLetters>> <Letters>> ' ';
  s.Size = ( <Tree <Mod <Dg 1> <Length <FuncNames>>> s.Size > );
  s.Choice s.Size =
    <Tree <Nth s.Choice <FuncNames>> ' _ ' <Nth s.Choice <FuncArity>>
      <- s.Size 1>>;
  s.FuncName ' _ ' 1 s.Size = s.FuncName ' _ ' 1 <Tree s.Size>;
  s.FuncName ' _ ' 2 0 = s.FuncName ' _ ' 2 <Tree 0> <Tree 0>;
  s.FuncName ' _ ' 2 s.Size = <Tree s.FuncName ' _ ' 2 <RandomSplit2 s.Size>>;
  s.FuncName ' _ ' 2 s.Size1 s.Size2 = s.FuncName ' _ ' 2 <Tree s.Size1> <Tree s.Size2>;
  s.FuncName ' _ ' 3 0 = s.FuncName ' _ ' 3 <Tree 0> <Tree 0> <Tree 0>;
  s.FuncName ' _ ' 3 s.Size = <Tree s.FuncName ' _ ' 3 <RandomSplit3 s.Size>>;
  s.FuncName ' _ ' 3 s.Size1 s.Size2 s.Size3 =
    s.FuncName ' _ ' 3 <Tree s.Size1> <Tree s.Size2> <Tree s.Size3>;
  s.FuncName ' _ ' 4 0 = s.FuncName ' _ ' 4 <Tree 0> <Tree 0> <Tree 0> <Tree 0>;
  s.FuncName ' _ ' 4 s.Size = <Tree s.FuncName ' _ ' 4 <RandomSplit4 s.Size>>;
  s.FuncName ' _ ' 4 s.Size1 s.Size2 s.Size3 s.Size4 =
    s.FuncName ' _ ' 4 <Tree s.Size1> <Tree s.Size2> <Tree s.Size3> <Tree s.Size4>;
}
```

# 1. Генерация выражения на Рефал-М

дерево-выражения {

0 = ' ' <выбранный-символ <остаток <сч> <число-букв>> <буквы>> ' ';

?размер = (<дерево-выражения <остаток <сч> <число-функций>> ?размер>);

?типкорня ?размер =

<дерево-выражения

<выбранный-символ ?типкорня <имена-функций> ' ' <выбранный-символ ?типкорня <арности-функций>>

<- ?размер 1>>;

?имяфункции ' ' 1 ?размер = ?имяфункции ' ' 1 <дерево-выражения ?размер>;

?имяфункции ' ' 2 0 = ?имяфункции ' ' 2 <дерево-выражения 0> <дерево-выражения 0>;

?имяфункции ' ' 2 ?размер = <дерево-выражения ?имяфункции ' ' 2 <случайное-разбиение-на-2 ?размер>>;

?имяфункции ' ' 2 ?размер1 ?размер2 =

?имяфункции ' ' 2 <дерево-выражения ?размер1> <дерево-выражения ?размер2>;

?имяфункции ' ' 3 0 = ?имяфункции ' ' 3 <дерево-выражения 0> <дерево-выражения 0> <дерево-выражения 0>;

?имяфункции ' ' 3 ?размер = <дерево-выражения ?имяфункции ' ' 3 <случайное-разбиение-на-3 ?размер>>;

?имяфункции ' ' 3 ?размер1 ?размер2 ?размер3 =

?имяфункции ' ' 3 <дерево-выражения ?размер1><дерево-выражения ?размер2><дерево-выражения ?размер3>;

?имяфункции ' ' 4 0 =

?имяфункции ' ' 4 <дерево-выражения 0> <дерево-выражения 0> <дерево-выражения 0> <дерево-выражения 0>;

?имяфункции ' ' 4 ?размер =

<дерево-выражения ?имяфункции ' ' 4 <случайное-разбиение-на-4 ?размер>>;

?имяфункции ' ' 4 ?размер1 ?размер2 ?размер3 ?размер4 =

?имяфункции ' ' 4

<дерево-выражения ?размер1> <дерево-выражения ?размер2>

<дерево-выражения ?размер3> <дерево-выражения ?размер4>;

}

# 1.Генерация выражений и таблиц истинности: промежуточные результаты

Содержимое файла выражений:

1 :  
(V \_ 2 (= \_ 2 B (= \_ 2 B C ))(& \_ 4 C C C (& \_ 3 B A A )))

2 :  
(V \_ 3 B (& \_ 3 C A (= \_ 2 C C ))(~ \_ 1 (V \_ 2 C C )))

3 :  
(& \_ 2 (V \_ 2 (& \_ 3 B A A )(V \_ 4 A C A C ))(& \_ 4 A A B B ))

4 :  
(V \_ 3 (V \_ 3 B C C )(V \_ 2 A A )(& \_ 2 C (& \_ 3 C B C )))

5 :  
(V \_ 3 (= \_ 2 (V \_ 4 B C B B ))(~ \_ 1 C )) A (& \_ 2 B A ))

...

Префиксная запись.

Конъюнкция: & \_ 2 двухместная, & \_ 3 трёхместная, & \_ 4 – четырёхместная

Дизъюнкция: V \_ 2 двухместная, V \_ 3 трёхместная, V \_ 4 – четырёхместная

Эквиваленция: = \_ 2

Импликация: > \_ 2

Отрицание: ~ \_ 1

# 1.Генерация выражений и таблиц истинности: промежуточные результаты

**Содержимое файла команд GNU Octave  
(для выражения №1):**

```
clear
NumberOfLetters=3 ;
TreeNum=1 ;
tablist_prepare
X111=B ;
T = tablist_addvar(T,X111,"X111 : B ");
X1121=B ;
T = tablist_addvar(T,X1121,"X1121 : B ");
X1122=C ;
T = tablist_addvar(T,X1122,"X1122 : C ");
X112=func_eq2(X1121,X1122);
T = tablist_addvar(T,X112,"X112 : X1121 =
X1122");
X11=func_eq2(X111,X112);
T = tablist_addvar(T,X11,"X11 : X111 = X112");
X121=C ;
T = tablist_addvar(T,X121,"X121 : C ");
X122=C ;
```

Листья дерева

Вычисление узла

```
T = tablist_addvar(T,X122,"X122 : C ");
X123=C ;
T = tablist_addvar(T,X123,"X123 : C ");
X1241=B ;
T = tablist_addvar(T,X1241,"X1241 : B ");
X1242=A ;
T = tablist_addvar(T,X1242,"X1242 : A ");
X1243=A ;
T = tablist_addvar(T,X1243,"X1243 : A ");
X124=func_and3(X1241,X1242,X1243);
T = tablist_addvar(T,X124,...
"X124 : X1241 & X1242 & X1243");
X12=func_and4(X121,X122,X123,X124);
T = tablist_addvar(T,X12,...
"X12 : X121 & X122 & X123 & X124");
X1=func_or2(X11,X12);
T = tablist_addvar(T,X1,"X1 : X11 V X12");
tablist_output
```

Вывод таблицы

Объём программы для 1000 выражений ~1Мбайт

# 1.Генерация выражений и таблиц истинности: промежуточные результаты

Содержимое файла таблиц истинности:

```
1:  
{  
A  
B  
C  
!  
}  
[0 0 0 0 1 1 1 1;0 0 1 1 0 0 1 1;0 1 0 1 0 1 0 1;0 1 0 1 0 1 0 1]  
2:  
{  
A  
B  
C  
!  
}  
[0 0 0 0 1 1 1 1;0 0 1 1 0 0 1 1;0 1 0 1 0 1 0 1;1 0 1 1 1 1 1 1]  
...
```

# 1.Генерация выражений и таблиц истинности: целостность данных

Выражение номер 5.184

$$(B \wedge (A \vee B \vee C)) \wedge B \wedge (A \vee A \vee B) = (B \vee A)$$

Выражение номер 5.185

$$B = \overline{(A \vee A \vee A \vee A)} \wedge \overline{C}$$

Выражение номер 5.186

$$B \wedge ((C \vee B \vee (B \vee C \vee A \vee C)) = (B \vee A))$$

Выражение номер 5.187

$$(C \wedge A \wedge B \wedge B) \wedge \overline{A \vee C \vee A \vee A} \wedge C \wedge (B \vee C \vee B \vee A)$$

29 ноября 2022 г. 09:38:46 МКТ

|           |   |   |   |   |   |   |   |   |
|-----------|---|---|---|---|---|---|---|---|
| A         | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| B         | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| C         | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| Результат | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

Краткая таблица истинности выражения 5.171

|           |   |   |   |   |   |   |   |   |
|-----------|---|---|---|---|---|---|---|---|
| A         | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| B         | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| C         | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| Результат | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

29 ноября 2022 г. 09:38:56 МКТ

Выражение номер 5.195

$$A \wedge ((A \vee B) = (C \vee C)) \wedge B \wedge (B \wedge B \wedge A)$$

Выражение номер 5.196

$$((C \wedge B) \wedge A \wedge A \wedge (B \rightarrow A)) \vee C \vee B \vee (C \wedge C)$$

Выражение номер 5.197

$$A \rightarrow ((C = (C \vee A)) \wedge C \wedge \overline{C})$$

Выражение номер 5.198

$$\overline{A} \wedge A \wedge (B = (C \vee C \vee C \vee C))$$

|           |   |   |   |   |   |   |   |   |
|-----------|---|---|---|---|---|---|---|---|
| A         | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| B         | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| C         | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| Результат | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |

Краткая таблица истинности выражения 5.180

|           |   |   |   |   |   |   |   |   |
|-----------|---|---|---|---|---|---|---|---|
| A         | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| B         | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| C         | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| Результат | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |

## 2.Отказ от стандартной реализации арифметики: первые шаги (арифметика.реф)

всецифры = "Z" "I" "2" "3" "4" "5" "6" "7" "8" "9";

\* для разнообразия нуль обозначен Z (zero), единица I

нуль {

= <нуль <всецифры>>;

?цифрануль .остальные = ?цифрануль;

}

один {

= <один <всецифры>>;

?цифрануль ?цифраодин .остальные = ?цифраодин;

}

предыдущаяцифра {

?цифра = <предыдущаяцифра ?цифра <всецифры>>;

?цифра .до ?предыдущ ?цифра .после = ?предыдущ;

?цифра ?цифра .между ?последняя = ?последняя;

}

...

(полностью см. <https://github.com/STrusov/refal-machine>, каталог Примеры)

## 2.Отказ от стандартной реализации арифметики: “абсолютная точность” при умножении и делении на степени 2

Сколько битов в 796 Тбайт?

Программа на Рефале:  $796 \cdot 2^{43} = 7001690045677568$

Инженерный калькулятор:  $7.00169004568e15$

Некоторые студенты:  $7.00169004568$

Другие студенты:  $7960000000000000$

...

**Разбор ошибок.**

## 2.Отказ от стандартной реализации арифметики: “абсолютная точность” при работе с системами счисления

Запись любого рационального числа в позиционной системе счисления с натуральным основанием  $q$  – периодическая дробь:

Введите число:

55,101

Введите основание:

9

61, (08155847734452048418808741305526337666016270231237)

Введите число:

55,101

Введите основание:

12

47,12(6640473675A3320237938B1770119A7A5699606AB3B294A90  
3557B7485461889B9842830A44BAA214165225B5108092712B8)

**“На что ты способен без онлайн-калькулятора?”**

# 3. Программа на Рефал-М – русскоязычный текст

```
tex: заголовокдокумента конецдокумента; ←  
представлениедробей: нормальноепредставление;  
начало =
```

Используются функции  
заголовокдокумента  
и конецдокумента  
из модуля tex

```
  <напечатать <заголовокдокумента  
    ("twocolumn")  
    ("Задачи к одиннадцатому практическому занятию.")>>  
  <обработка <ввод>>  
  <напечатать <конецдокумента>>;
```

Опущены {}

```
обработка {  
  .номер ":" =  
  <напечатать .номер ". Вычислите $">  
  <обработка <ввод>>;  
  "А " .число =  
  <напечатать <нормальноепредставление .число> "_{2} \cdot ">  
  <обработка <ввод>>;  
  "В " .число =  
  <напечатать <нормальноепредставление .число> "_{2}$\n">  
  <обработка <ввод>>;  
  = ;
```

```
}  
ввод = <Card>;  
напечатать { .чтонибудь = <Prout .чтонибудь>; }
```

# 3. Программа на Рефал-М – русскоязычный текст

tex: заголовокдокумента конецдокумента;  
представлениедробей: нормальноепредставление;  
начало =

```
<напечатать <заголовокдокумента  
  ("twocolumn")  
  ("Задачи к одиннадцатому практическому занятию.")>>  
<обработка <ввод>>  
<напечатать <конецдокумента>>;
```

обработка {

```
  .номер ":" =  
  <напечатать .номер ". Вычислите $">  
  <обработка <ввод>>;  
  "А " .число =  
  <напечатать <нормальноепредставление .число> "_{2} \cdot ">  
  <обработка <ввод>>;  
  "В " .число =  
  <напечатать <нормальноепредставление .число> "_{2}$\n">  
  <обработка <ввод>>;  
  = ;
```

}

ввод = <Card>;

напечатать { .чтонибудь = <Prout .чтонибудь>; }

**Как грамматически согласовать имена функций?**

# 3. Программа на Рефал-М – русскоязычный текст

tex: заголовокдокумента конецдокумента;  
представлениедробей: нормальноепредставление;  
начало =

```
<напечатать <заголовокдокумента  
  ("twocolumn")  
  ("Задачи к одиннадцатому практическому занятию.")>>  
<обработка <ввод>>  
<напечатать <конецдокумента>>;  
обработка {  
  .номер ":" =  
  <напечатать .номер " . Вычислите $">  
  <обработка <ввод>>;  
  "А " .число =  
  <напечатать <нормальноепредставление .число> "_{2} \cdot ">  
  <обработка <ввод>>;  
  "В " .число =  
  <напечатать <нормальноепредставление .число> "_{2}$\n">  
  <обработка <ввод>>;  
  = ;  
}  
ввод = <Card>;  
напечатать { .чтонибудь = <Prout .чтонибудь>; }
```

<печатать ...>  
Возникает проблема  
с падежами аргументов.

# 3. Программа на Рефал-М – русскоязычный текст

tex: заголовокдокумента конецдокумента;  
представлениедробей: нормальноепредставление;  
начало =

```
<напечатать <заголовокдокумента  
  ("twocolumn")  
  ("Задачи к одиннадцатому практическому занятию.")>>
```

<обработать ...>

```
<обработка <ввод>>
```

```
<напечатать <конецдокумента>>;
```

обработка {

```
.номер ":" =
```

```
<напечатать .номер ". Вычислите $">
```

```
<обработка <ввод>>;
```

```
"А " .число =
```

```
<напечатать <нормальноепредставление .число> "_{2} \cdot ">
```

```
<обработка <ввод>>;
```

```
"В " .число =
```

```
<напечатать <нормальноепредставление .число> "_{2}$\n">
```

```
<обработка <ввод>>;
```

```
= ;
```

```
}
```

```
ввод = <Card>;
```

```
напечатать { .чтонибудь = <Prout .чтонибудь>; }
```

# 3. Программа на Рефал-М – русскоязычный текст

tex: заголовокдокумента конецдокумента;  
представлениедробей: нормальноепредставление;  
начало =

```
<напечатать <заголовокдокумента  
  ("twocolumn")  
  ("Задачи к одиннадцатому практическому занятию.")>>  
<обработать <ввод>>  
<напечатать <конецдокумента>>:  
обработать {  
  .номер ":" =  
  <напечатать .номер . вычислите  $\varphi$  >  
  <обработать <ввод>>;  
  "А " .число =  
  <напечатать <нормальноепредставление .число> "_{2} \cdot ">  
  <обработать <ввод>>;  
  "В " .число =  
  <напечатать <нормальноепредставление .число> "_{2}$\n">  
  <обработать <ввод>>;  
  = ;  
}
```

Использование глаголов уместно для функций, вызываемых ради побочных эффектов, а не для возвращаемых значений.

```
ввод = <Card>;  
напечатать { .чтонибудь = <Prout .чтонибудь>; }
```

# 3. Программа на Рефал-М – русскоязычный текст

дерево-выражения {

0 = ' ' <выбранный-символ <остаток <сч><число-букв>> <буквы>> ' ';

?размер =

(<дерево-выражения <остаток <сч><число-функций>> ?размер>)

?типкорня ?размер =

<дерево-выражения

<выбранный-символ ?типкорня <имена-функций> ' \_ ' <выбранный-символ ?типкорня <арности-функций>>

<- ?размер 1>>;

?имяфункции ' \_ ' 1 ?размер = ?имяфункции ' \_ ' 1 <дерево-выражения ?размер>;

?имяфункции ' \_ ' 2 0 = ?имяфункции ' \_ ' 2 <дерево-выражения 0><дерево-выражения 0>;

?имяфункции ' \_ ' 2 ?размер = <дерево-выражения ?имяфункции ' \_ ' 2 <случайное-разбиение-на-2 ?размер>>;

?имяфункции ' \_ ' 2 ?размер1 ?размер2 =

?имяфункции ' \_ ' 2 <дерево-выражения ?размер1><дерево-выражения ?размер2>;

?имяфункции ' \_ ' 3 0 = ?имяфункции ' \_ ' 3 <дерево-выражения 0><дерево-выражения 0><дерево-выражения 0>;

?имяфункции ' \_ ' 3 ?размер = <дерево-выражения ?имяфункции ' \_ ' 3 <случайное-разбиение-на-3 ?размер>>;

?имяфункции ' \_ ' 3 ?размер1 ?размер2 ?размер3 =

?имяфункции ' \_ ' 3 <дерево-выражения ?размер1><дерево-выражения ?размер2><дерево-выражения ?размер3>;

?имяфункции ' \_ ' 4 0 =

?имяфункции ' \_ ' 4 <дерево-выражения 0><дерево-выражения 0><дерево-выражения 0><дерево-выражения 0>;

?имяфункции ' \_ ' 4 ?размер =

<дерево-выражения ?имяфункции ' \_ ' 4 <случайное-разбиение-на-4 ?размер>>;

?имяфункции ' \_ ' 4 ?размер1 ?размер2 ?размер3 ?размер4 =

?имяфункции ' \_ ' 4

<дерево-выражения ?размер1><дерево-выражения ?размер2>

<дерево-выражения ?размер3><дерево-выражения ?размер4>;

}

Использование существительных уместно, если функция нужна, чтобы возвращать значение.

## 4.Выводы

Автоматическая массовая генерация примеров и ответов

- позволяет преподавателю прояснить понимание формальных систем,
- обеспечивает преподавателя материалом для иллюстраций и проверки, но не избавляет от необходимости ставить интересные задачи.

Отказ от стандартной реализации арифметики позволяет

- производить разбор ошибок, совершаемых студентами (или – если студент доказал правильность своего решения – самим преподавателем при написании программы)
- решать задачи, для которых (пока что?) нет онлайн-калькуляторов.

Связка “Рефал + TeX” пригодна для решения задач, связанных с автоматической генерацией, преобразованиями и документированием выражений, текстов программ, таблиц и другой структурированной информации.

Для генерации псевдослучайных чисел и работы с массивами лучше использовать вычислители программ в структурном стиле.

Использование Рефал-М

- позволяет писать программы, читаемые для русскоязычного пользователя
- более остро ставит вопросы о стиле написания и понимании текста программ.

## 5.Перспективы и нерешённые вопросы

### **Перспективы:**

- Учить не только использовать формальные правила, но и думать
- Учить эвристическим приёмам, позволяющим проявить интуицию и мышление
- Использовать опыт в преподавании дисциплин “Микропроцессорные системы”, “Мастер по обработке цифровой информации” (специальность “Компьютерные системы и комплексы”)?
- Использовать связку “Рефал-М + TeX” для решения других задач (преобразования документов?)

### **Нерешённые вопросы:**

- Формализация и автоматическая генерация алгоритмов и блок-схем?

**Спасибо за внимание!**

Даниил Исакевич,  
daniil.isakevitch@yandex.ru  
+7-905-140-10-16