

Методы отладки функциональных программ

В. Н. Касьянов, Е. В. Касьянова, К. А. Кламбоцкий
*Институт систем информатики
им. А.П. Ершова СО РАН*



В докладе рассматриваются методы отладки функциональных программ и опыт их использования для отладки Cloud Sisal программ в рамках создаваемой в ИСИ СО РАН системы CPPS поддержки облачного параллельного программирования.

Система CPPS

Перемножение квадратных матриц [alexm](#) [logout](#)

Project/Module Settings Modules: Projects:

Module settings:

Module name:

Module description:

Project Settings:

Input data:

```
1 {  
2   "A": [  
3     [1,2,2,1],  
4     [3,4,4,1],  
5     [1,2,1,1],  
6     [3,4,4,1]  
7   ],  
8  
9   "B": [  
10    [4,3,3,1],  
11    [2,1,1,1],  
12    [4,3,1,1],  
13    [4,3,1,1]  
14  ],  
15  
16   "N": 4  
17 }
```

Project name:

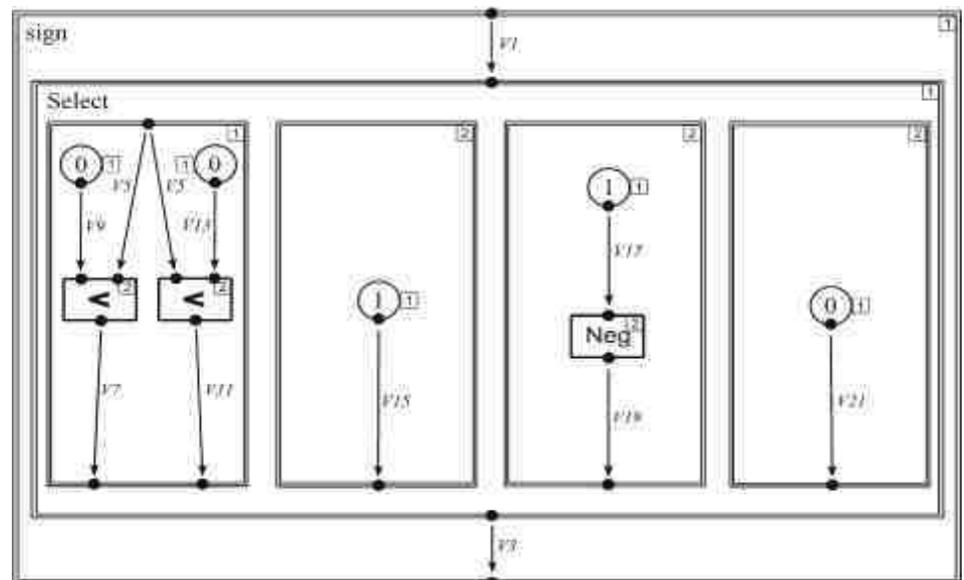
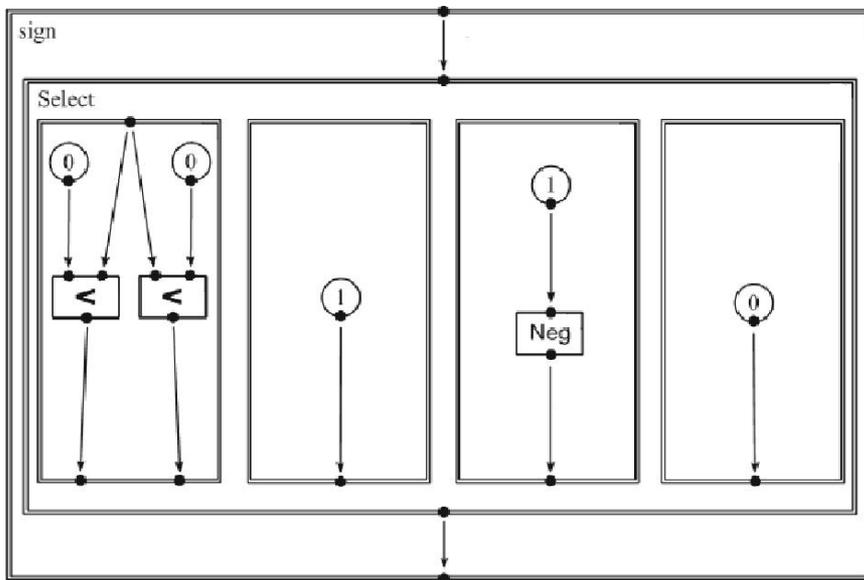
Project description:

main

```
1 definition main  
2  
3 function main(A,B: array[array[integer]]; N : integer returns array[array[integer]] )  
4   for i in 1, N  
5     returns array of  
6       for j in 1, N  
7         returns array of  
8           for k in 1, N  
9             returns sum of A[i,k] * B[k,j]  
10          end for  
11        end for  
12      end for  
13 end function  
14
```

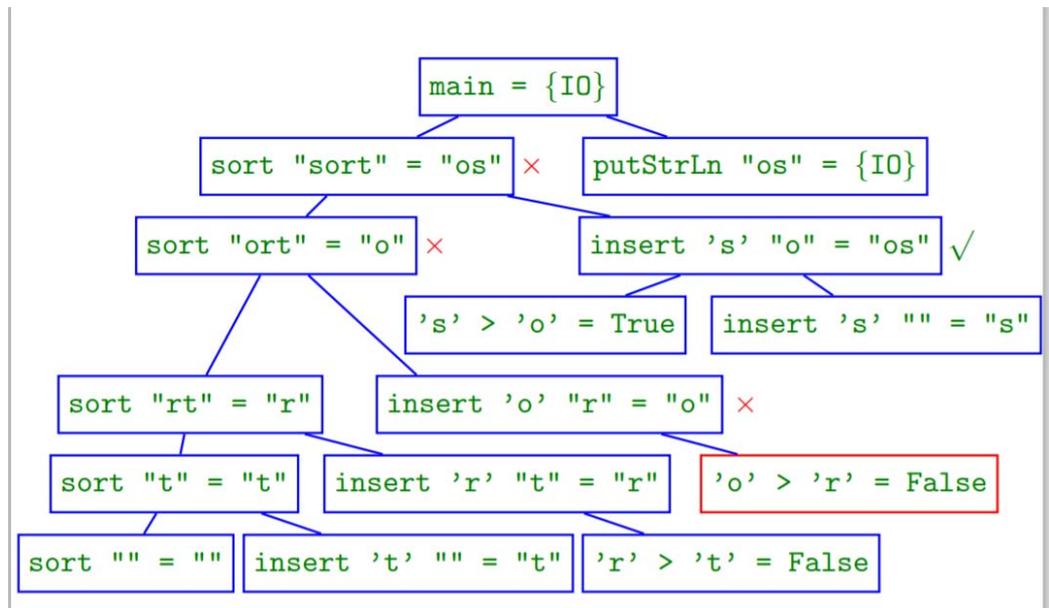
Язык Cloud Sisal и его графовое представление

function sign (N: integer returns integer)
 if N > 0 then 1
 elseif N < 0 then -1 else 0 end if
end function

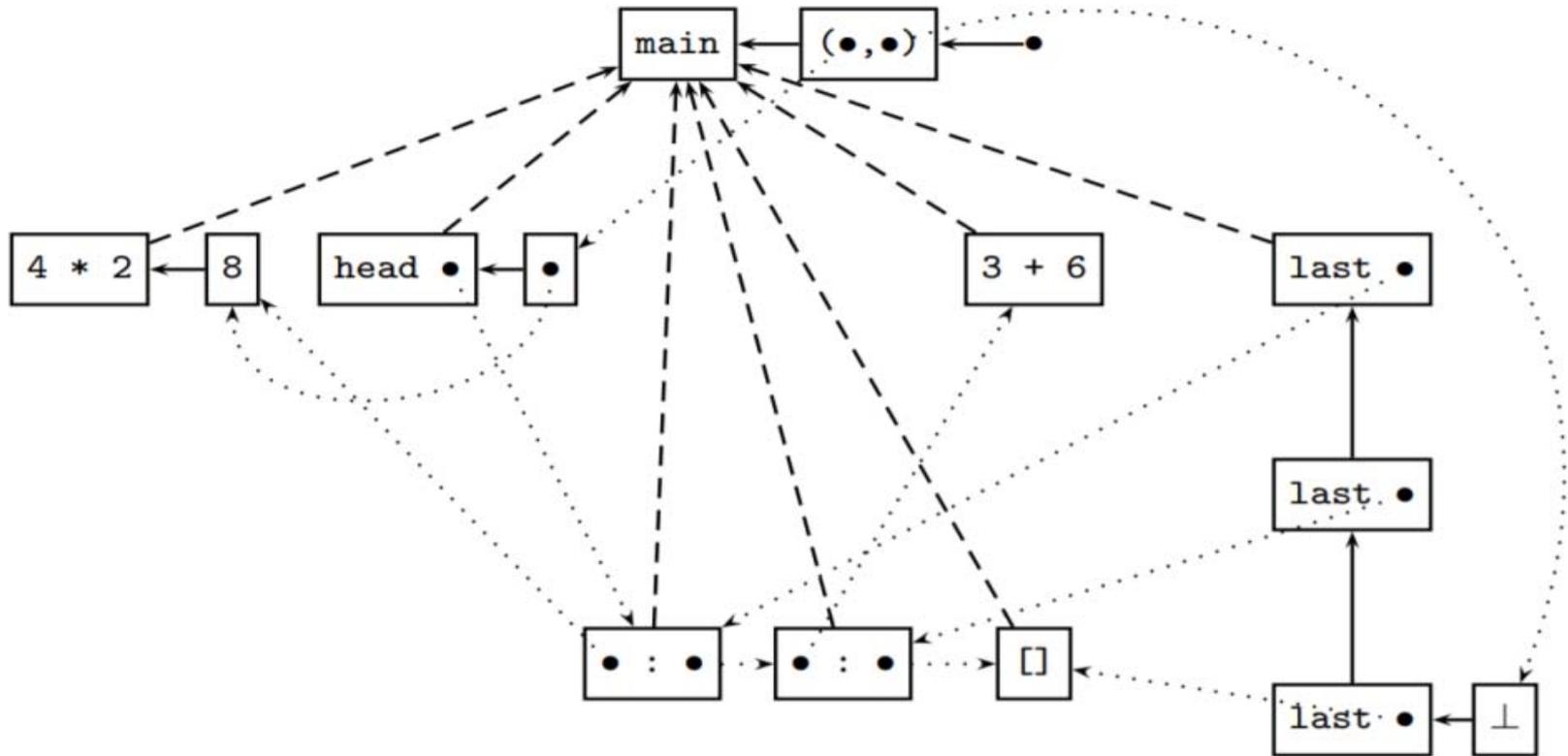


Методы отладки для языков Haskell, F# и Common Lisp

sort "sort" = "os" ?	NO
insert 's' "o" = "os" ?	YES
sort "ort" = "o" ?	NO
insert 'o' "r" = "o" ?	NO



Redex Trail graph



След функции (factorial 2)

```
(trace factorial)

(factorial 2)
  0: (FACTORIAL 3)
    1: (FACTORIAL 2)
      2: (FACTORIAL 1)
        3: (FACTORIAL 0)
          3: FACTORIAL returned 1
        2: FACTORIAL returned 1
      1: FACTORIAL returned 2
    0: FACTORIAL returned 6
6

(untrace factorial)
```

Заключение

В ходе выполнения данной работы были изучены специальные методы отладки функциональных программ на языках Haskell, F# и Common Lisp и из них были выбраны для отладки Cloud Sisal программ продвинутые методы отладки языка Haskell: Freja — интерактивное раскрытие редексов, Hood — слежение за изменением данных, и Redex Trail System — визуализация дерева раскрытия редексов. На их основе были разработаны методы отладки Cloud Sisal программ и выполнена их экспериментальная реализация для представительного подмножества языка Cloud Sisal.

Предложенные в работе методы представляют интерес для применения в системе CPPS, поскольку они дополняют используемые в системе методы визуальной отладки Cloud Sisal программ на основе их графового представления [2, 4] и опираются на успешную практику применения подобных методов для других функциональных языков.

Спасибо !