

Алгоритм прогонки вызовов рекурсивных функций, не приводящий к зацикливанию в компиляторе Рефала-5λ

М. Д. Апахов

МГТУ имени Н. Э. Баумана

IV совместное рабочее совещание ИПС имени А.К. Айламазяна РАН и
МГТУ имени Н.Э. Баумана по функциональному языку программирования

Рефал

8 июня 2021 года

Существующий алгоритм прогонки. Суть

Суть прогонки — объединение нескольких шагов выполнения программы в один

Результат прогонки — набор решений

Решение — новое выражение E и набор сужений C_1, \dots, C_N

Недостаток — итеративный подход, необходимый для построения прогонки сложного выражения

Существующий алгоритм прогонки. Примеры

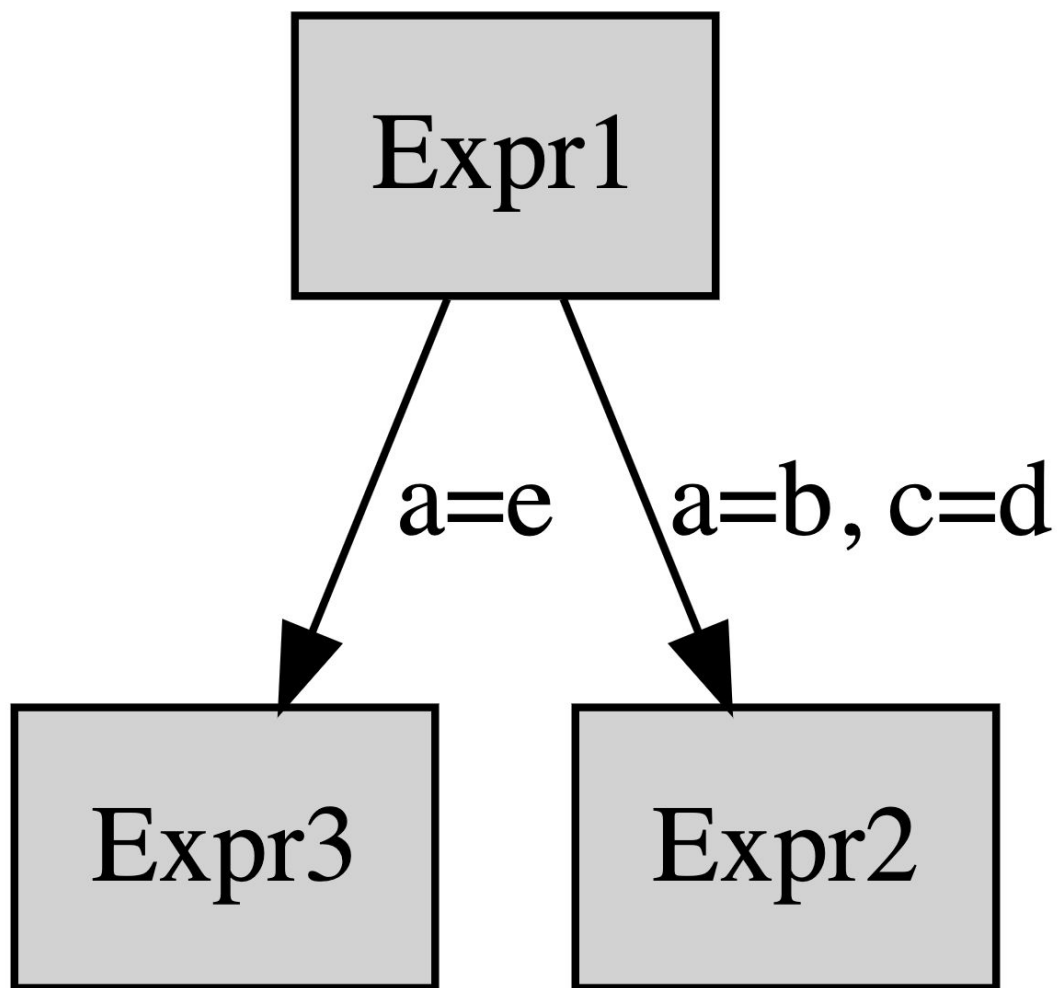
```
$INLINE Apply;
Apply {
    s.Fn e.Argument = <s.Fn e.Argument>;
    (t.Closure e.Bounded) e.Argument
        = <Apply t.Closure e.Bounded e.Argument>;
}
F { e.Args = <Apply Foo e.Args>}

/* после встраивания */
F { e.Args = <Foo e.Args>}
```

Существующий алгоритм прогонки. Примеры

```
$INLINE Loop;  
Loop {  
    e.X = <Loop e.X A>;  
}  
  
$ENTRY Go {  
    = <Exit 0> <Loop>;  
}
```

Новый алгоритм прогонки. Один шаг прогонки

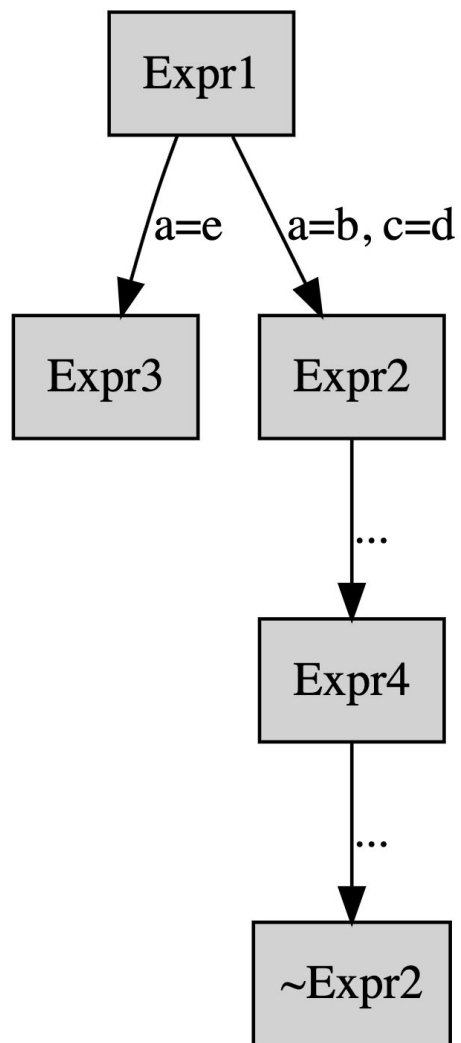


Новый алгоритм прогонки. Суть

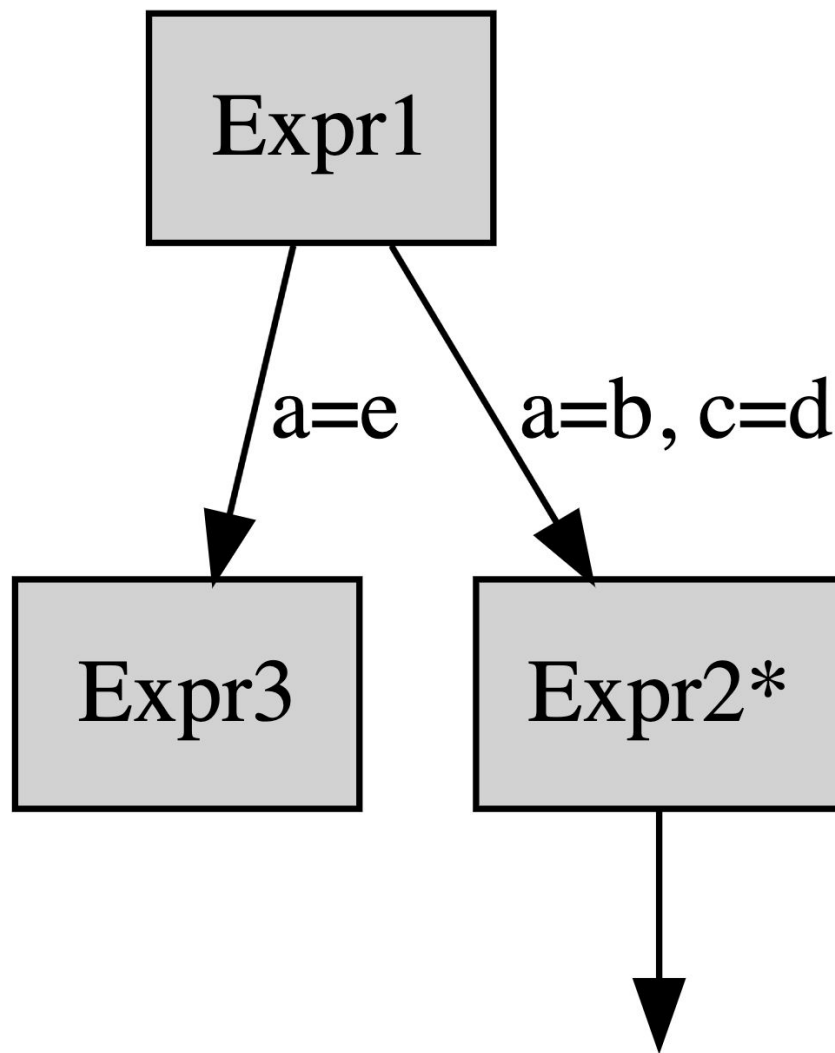
Вместо остановки на первом шаге, продолжаем прогонять получившиеся выражения.

При нахождении похожих конфигураций, производим перестройку сверху — замораживаем вызов в предке, удаляем всех потомков, делаем попытку прогнать снова

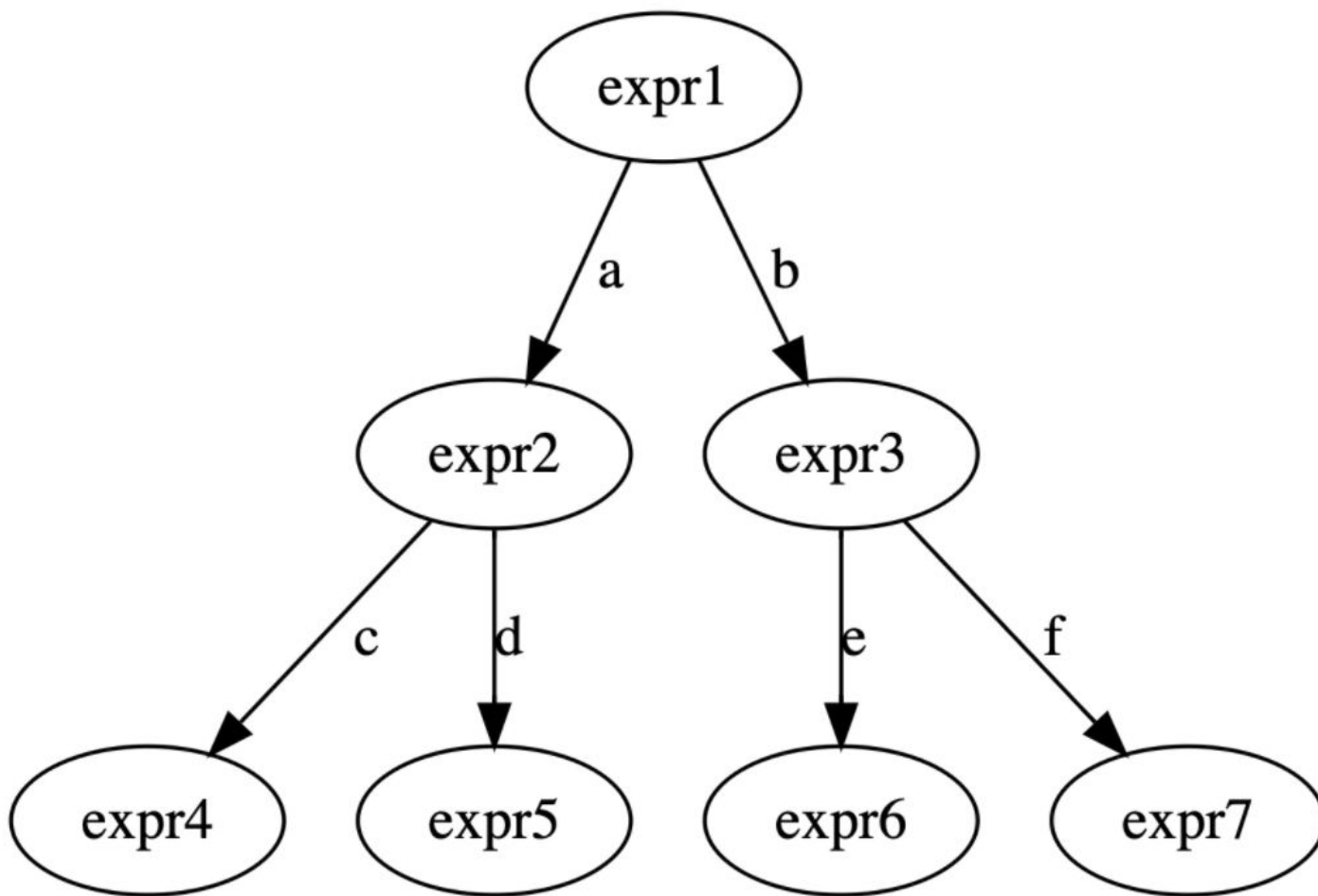
Новый алгоритм прогонки. Построение графа



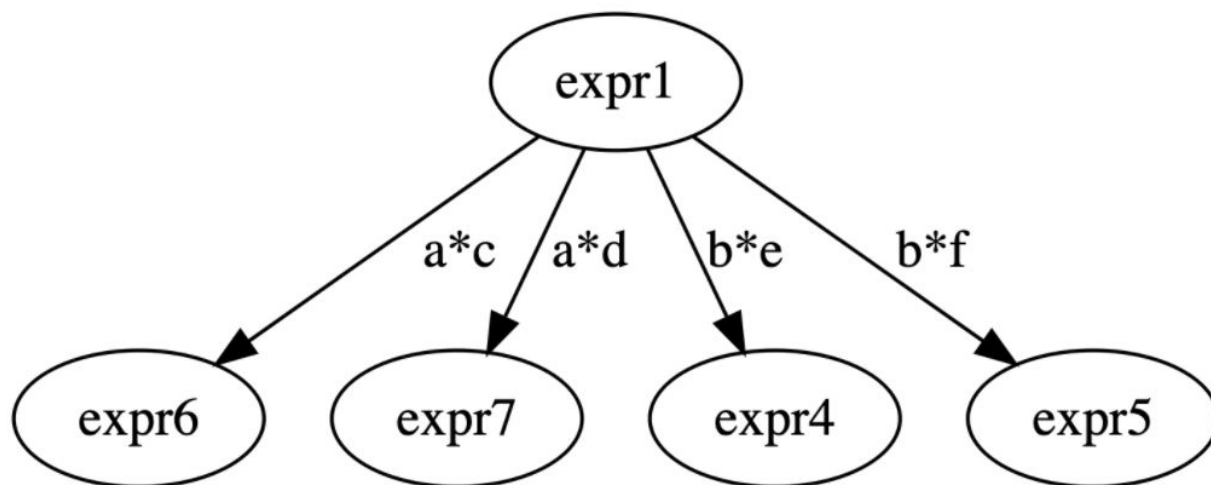
Новый алгоритм прогонки. Построение графа



Новый алгоритм прогонки. Граф конфигураций



Новый алгоритм прогонки. Приведение к результату прогонки



Новый алгоритм прогонки. Примеры

```
$INLINE Loop;  
Loop {  
    e.X = <Loop e.X A>;  
}  
  
$ENTRY Go {  
    = <Exit 0> <Loop>;  
}
```

Новый алгоритм прогонки. Примеры

```
$DRIVE ADD;  
ADD {  
    (Z) (e.X) = e.X;  
    ('1' e.N) (e.X) = '1' <ADD (e.N) (e.X)>;  
}  
  
TRIADD {  
    (e.A) (e.B) (e.C) = <ADD (<ADD (e.A) (e.B)>)  
    (e.C)>;  
}  
  
$ENTRY Go {  
    = <Prout <ADD ('11' Z) ('111' Z)>>;  
}
```

Новый алгоритм прогонки. Примеры

```
$DRIVE INC;  
ADD {  
    (t.X) = <Add t.X 1>;  
}  
  
$ENTRY Go {  
    = <Prout <MAP &INC (1) (2) (3)>>;  
}  
  
$DRIVE MAP;  
MAP {  
    t.Closure t.Elem e.Rest =  
        <t.Closure t.Elem> <MAP t.Closure e.Rest>;  
    t.Closure =;  
}
```

Новый алгоритм прогонки. Преимущества

Один цикл прогонки для каждого вызова

Нет различий между \$DRIVE и \$INLINE

Все функции безопасны для прогонки

Заключение