

Документация към проект
Интерпретатор на *Prolog*
ФП, зимен семестър 2022/2023

Калоян Цветков
kaloyants25@gmail.com

ФМИ, СУ
1.0.0

Съдържание

1	Кратко описание	2
2	Структура	2
3	Основни концепции и функции	2
3.1	Унификация на термове	2
3.2	Резолюция	3
4	Примерна работа	3

1 Кратко описание

Интерпретаторът е написан на езика Haskell и компилиран чрез GHC, версия 8.10.7.

Може да определя истинността на факти, заредени от програма на езика *Prolog*, използвайки директно записаните в него константни факти или използвайки изводи от правила.

При подаване на факт, съдържащ променлива (заявка), интерпретаторът определя всички стойности на променливата, за които фактът е верен (ако има такива според кода) и при поискване ги принтира. Също така може да разрешава равенства на термове, отново посочвайки за кои стойности на променливите равенството е изпълнено.

2 Структура

Сорс-кодът е разделен в 7 модула:

- Datatypes
Съдържа всички потребителски типове, представляващи елементите на една програма на *Prolog*;
- Checkers
Предикати, определящи дали низ може да се конвертира до ??? семантичен елемент от *Prolog* и други;
- Conversions
Функции за конвертиране между ??? семантични елементи на *Prolog*;
- Identities
Функции за проверка на идентичност на потребителски типове;
- Tools
Общи функции за обработка на данни;
- Unification
Реализация на алгоритъма за унификация на термове
- Resolutions
Основен метод; съдържа функции за проверка на истинност на факти.

3 Основни концепции и функции

3.1 Унификация на термове

За унификация на два терма се грижи функцията

```
toBeUnified :: (Term, Term) -> QueryResult
```

Функцията имплементира следния алгоритъм за унификация:

```
Input: Terms A, B
  Initialize the result list to be empty;
  Initialize the stack with pair (A,B);
  Initialize the fail flag to false;
  While stack is not empty and not fail do:
    pop (X,Y) from the stack
    Case
      X is a variable:
```

1
2
3
4
5
6
7
8

```

        Substitute X with Y in the stack
        add X=Y to the list
Y is a variable:
    Substitute Y with X in the stack
    add Y=X int the list
X and Y are identical constants
    continue
X is atom f(X_1,...,X_n) and Y is atom f(Y_1,...,Y_n) for some identifier
    f and n>0:
        for i=1..n: push X_i=Y_i in the stack
otherwise:
    set fail to true
end Case
end While
Output:
    If fail, then empty QueueResult
    else QueryResult, corresponding to the result list;

```

3.2 Резолюция

4 Примерна работа

Входен файл: *prolog/factsandrules.pl*:

```

%father/2
father(ruwee, padme).
father(anakin, luke).
father(anakin, leia).
father(han, ben).

%mother/2
mother(jobal, padme).
mother(shmi, anakin).
mother(padme, luke).
mother(padme, leia).
mother(leia, ben).

%alias/2
alias(darthvader, anakin).
alias(kyloren, ben).
alias(X,Y) :- alias(Y,X).

%parent/2
parent(X,Y) :- father(X, Y).
parent(X,Y) :- mother(X, Y).

%parent/2
childof(X,Y) :- parent(Y,X).

```

Стартиране на интерпретатора (в конзолен режим):

```
Which file to consult from the directory "prolog/"?
```

```
> factsandrules.pl
true.
> father(anakin,luke).
true.
> father(anakin,luke)
You are allowed to input only facts, queries and equalities!
> father(luke,anakin).
false.
> father(Parent,Child).
Parent = ruwee.
Child = padme.

Parent = anakin.
Child = luke.

Parent = anakin.
Child = leia.

Parent = han.
Child = ben.

You are allowed to input only facts, queries and equalities!
> a(X)=a(arg).
X = arg.
> a(Y,r)=b(X,Z).
false.
> a(Y,r)=a(X,Z).
Y = X.
Z = r.
> pred(Z,Z)=pred(X,t).
X = t.
Z = t.
> pred(Z,Z)=pred(X,Y).
Y = X.
Z = Y.
> quit
Consult another file? ( y | [n] )
> no
Closing...
```
