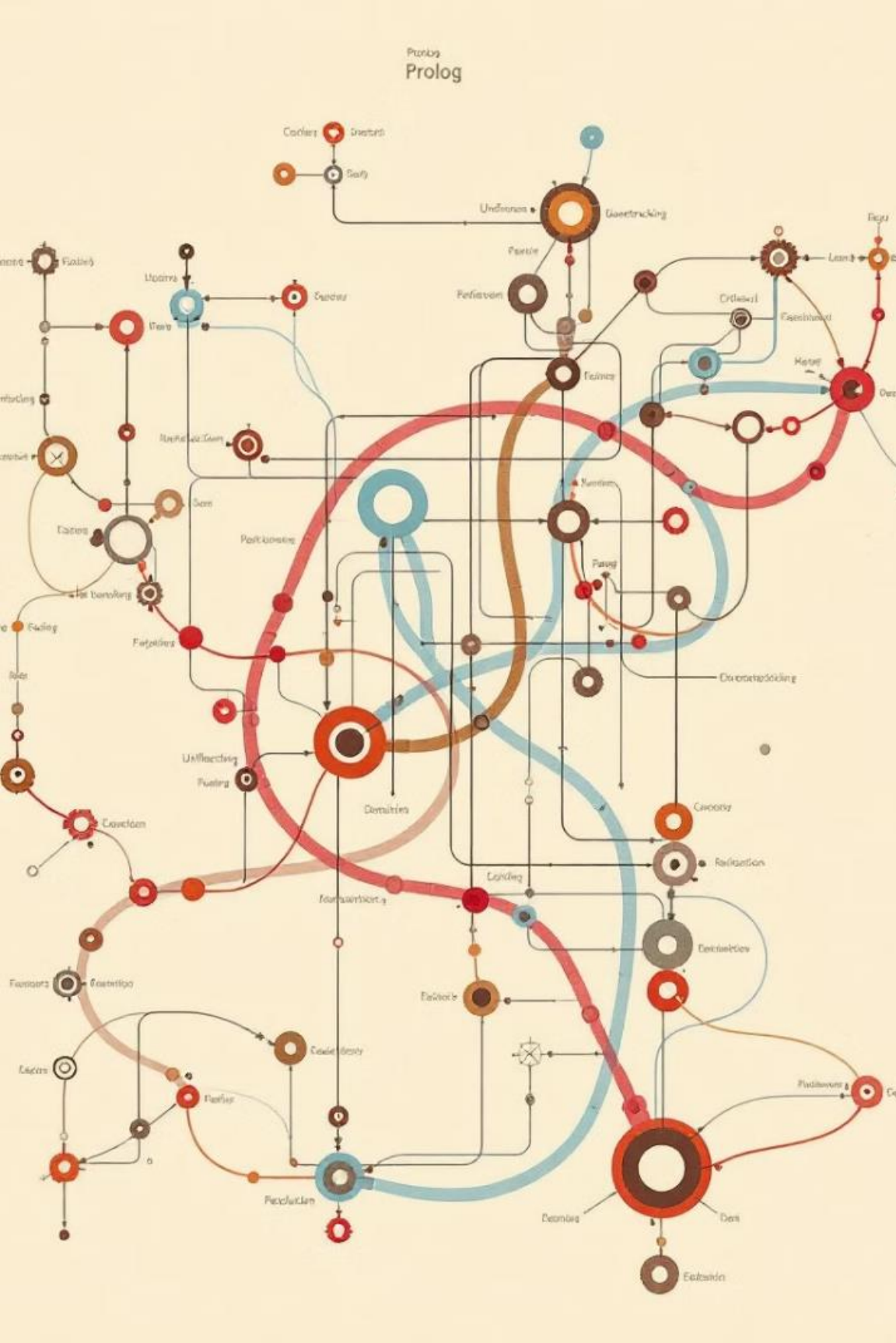


Семантика на езика Пролог.  
Унификация.  
Вградени предикати



# Семантика на Пролог

1

## Унификация

Съпоставяне на термове.

2

## Проследяване

Търсене на решения.

(3)

Връщане назад

Отказ от неуспешни опити.

# Унификация в Пролог

Унификацията е сърцето на Пролог. Тя определя как променливите се свързват. Също така, как се съпоставят терми. Примерите показват как работи този процес.

## Променливи

Променливите могат да бъдат свободни или свързани.

## Термини

Терми са константи, променливи или структури.

# Унификация

Да разгледаме формално процеса на унификация. Нека са дадени два терма  $T1$  и  $T2$ . Пролог прави унификация по следния начин:

- Ако  $T1$  и  $T2$  са атоми или числа, тогава те се унифицират когато са идентични;
- Ако  $T1$  е свободна променлива, тогава тези терми се унифицират като  $T1$  се свързва с  $T2$ ;
- Ако  $T2$  е свободна променлива, тогава се унифицират като  $T2$  се свързва с  $T1$ ;
- **Ако  $T1$  и  $T2$  са структури, тогава те се унифицират когато:**
  - $T1$  и  $T2$  притежават еднакви имена;
  - Имат еднакъв брой аргументи;
  - Съответните двойки аргументи унифицират.

За да демонстрираме унификацията в Пролог, ще разгледаме отново примерната база данни на сюжета от „Железният светилник“.

мъж(стоян). мъж(кочо). мъж(лазар).

жена(султана). жена(манда).

жена(нона). жена(катерина).

родител(стоян,кочо).

родител(стоян,лазар).

родител(стоян,манда).

родител(стоян,нона).

родител(стоян,катерина).

родител(султана,кочо).

родител(султана,лазар).

родител(султана,манда).

родител(султана,нона).

родител(султана,катерина).

баща(X) :- родител(X, \_), мъж(X).

майка(X) :- родител(X, \_), жена(X).

сестра(X,Y):-родител(Z,X), родител(Z,Y),

жена(X), жена(Y), X \== Y.

## Важни моменти при осъществяването на унификация:

- Когато Пролог се опита да удовлетвори целта, търсенето на съвпадение започва от началото на програмата.
- Когато се прави ново обръщение за удовлетворяване, търсенето на съвпадение отново започва от началото на програмата.
- Когато при търсене на съвпадение се стигне до успех, започва да се търси решение за следващата подцел.
- Когато една променлива бъде обвързана в клауза единствения начин за нейното освобождаване е чрез механизма на възврат.

## Унификация на сложни обекти

Сложните обекти могат да се унифицират както с отделна променлива, така и със сложен обект. Това означава, че е възможно да се използва сложния обект като едно цяло, а също да се използва унификация на отделните елементи. Например: `date(14,2,2023)`. Ако се съпостави с отделна променлива `X` на този обект, то като цяло променливата `X` ще се обвърже с `date(14,2,2023)`. От друга страна е възможно `date(14,2,2023)` да се съпостави на `date(DD,MM,YYYY)` и по този начин обвързването на променливите да е следното:

`DD` с 14, `MM` с 2 и `YY` с 2023.



## Използване на знак за равенство за унификация на сложни обекти

Пролог прилага унификация по два начина. Първият е при **удовлетворяване на цел с глава на клауза**. Вторият начин **преминава през знака за равенство**, който на практика е инфиксен предикат, т.е. предикат, който се намира между аргументите си, а не преди тях. Пролог извършва необходимите съпоставяния на обектите от двете страни на равенството. Това се използва за намиране на стойностите на аргументи в сложен тип обект.



## Унифициращ алгоритъм. Пример.

Терми. Съставни терми (структури) – представят съставни обекти (не функция, която връща резултат). Записване:  $f(a_1, a_2, a_3, \dots, a_n)$ ; където  $f$  е функтор, а термите  $a_1, a_2, a_3, \dots, a_n$  са аргументи. Терми са също променливите и константите.

Пример 1: `owns(tom, horse(blackly))`. В този пример структурата `horse(blackly)` представя кон с неговото име. `horse` е функтора, а името `blackly` е аргумента.

Пример 2: `owns(tom, car(mercedes, red))`. Структурата `car(mercedes, red)` представя вида и цвета на колата. Функтурът е `car`, а аргументите са два – `mercedes` и `red`.

## Процесът на унификация на терми:

- *Константа* (атомен терм) се унифицира с равна на нея константа, свободна променлива или свързана променлива стойност, равна на константата;
- *Променлива* – ако е свободна се унифицира с всеки терм, ако е свързана се унифицира по правилото за константи или съставни терми, в зависимост от стойността ѝ;
- *Съставен терм* се унифицира само с такъв съставен терм, който има същия функтор и същия брой аргументи, и на който аргументите се унифицират със съответните аргументи на дадения терм.

Унификацията служи за:

- Присвояване (свързване на променлива със стойност);
- Предаване на параметри;
- Проверка на равенство;
- Достъп до структури от данни и техните елементи.

**Пример:** Нека са дадени предикатите:

book(title, pages).

written\_by(author, title).

long\_novel(title).

И следната програма:

written\_by(димитър\_талев, железният\_светилник).

written\_by(елин\_пелин, гераците).

book(гераците, 112 ).

book(железният\_светилник, 498).

long\_novel(Title):-

    written\_by(\_, Title),

    book(Title, Length),

    Length > 300.

Нека проследим процеса при вземане на решения от Пролог.

## Първа цел: `written_by(X, Y)`.

Когато Пролог се опитва да удовлетвори целта `written_by(X, Y)`, е необходимо да се тестват всички клаузи в програмата за съвпадение.

При опита за съвпадение на променливите X и Y с аргументите на всяка `written_by` клауза, Пролог ще започне търсенето от началото на програмата към нейния край.

written\_by( X , Y ).

written\_by(димитър\_талев, железният\_светилник)

Пролог намира съвпадение и X обвързва с димитър\_талев, а Y се обвързва с железният\_светилник, и извежда следния резултат:

X=димитър\_талев, Y=железният\_светилник

Тъй като се проверява за всички възможни решения на целта, Пролог унифицира свободните променливи и с втората клауза на written\_by:

written\_by(елин\_пелин, гераците).

След това извежда и второто решение:

X= елин\_пелин, Y=гераците



```
1 written_by(димитър_талев,железният_светилник).  
2 written_by(елин_пелин,гераците).  
3 book(гераците,112 ).  
4 book(железният_светилник, 498).  
5 long_novel(Title):- written(_,Title),book(Title,Length),Length > 300.  
6
```



*written\_by*(X,  
Y).



**X** = димитър\_талев,

**Y** = железният\_светилник

**X** = елин\_пелин,

**Y** = гераците

?-

written\_by(X, Y).|

## Втора цел: written\_by (X, гераците).

Пролог ще опита да намери съвпадение с първата клауза `written_by`:

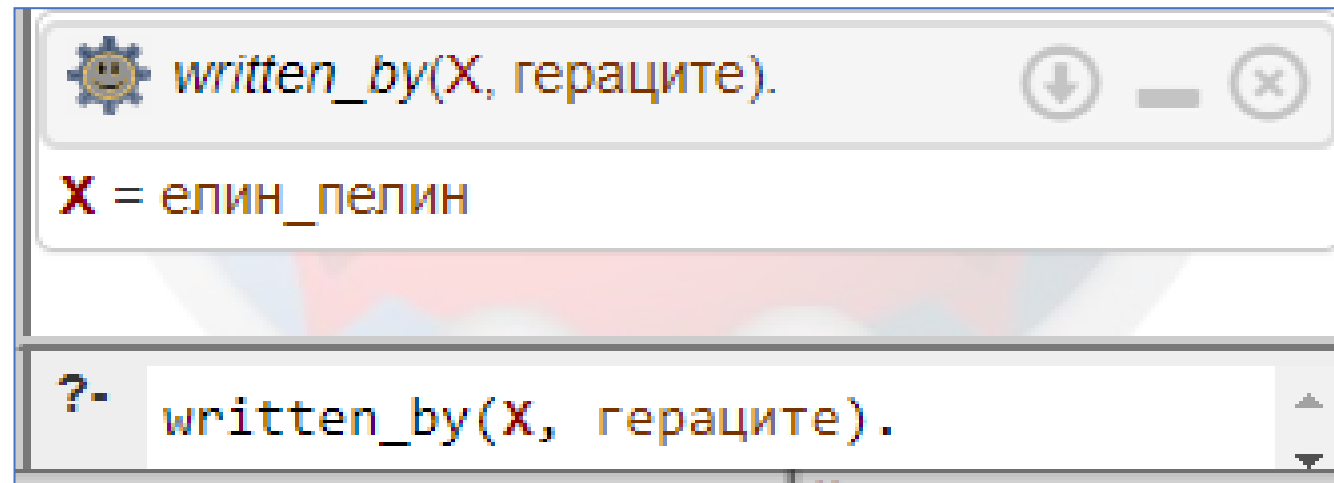
written\_by ( X , гераците ).

written\_by(димитър\_талеv, железният\_светилник).

Тъй като „гераците“ и „железният светилник“ не си съвпадат, опитът за унификация пропада. Пролог ще продължи със следващите факти в програмата:

written\_by(елин\_пелин, гераците).

Тук ще се намери унификация и X ще се обвърже с „елин\_пелин“.





**Трета цел:** `long_novel(X)`.

Пролог се опитва да удовлетвори целта, като се изследва дали целта не съвпада с някой факт или с глава на правило.

В този случай се намира съвпадение с:  
`long_novel(Title)`

-----

Целта съвпада с глава на правило и унификацията е направена. Пролог ще прави опит за удовлетворяване на подцелите на това правило.

```
long_novel(Title):- written_by(_, Title),  
book(Title, Length), Length>300.
```

Удовлетворяването на тялото на правилото ще премине първо през първата подцел `written_by(_, Title)`. Трябва да се обърне внимание, че авторът на книгата не е от значение, тъй като на мястото на `author` е поставена анонимна променлива.

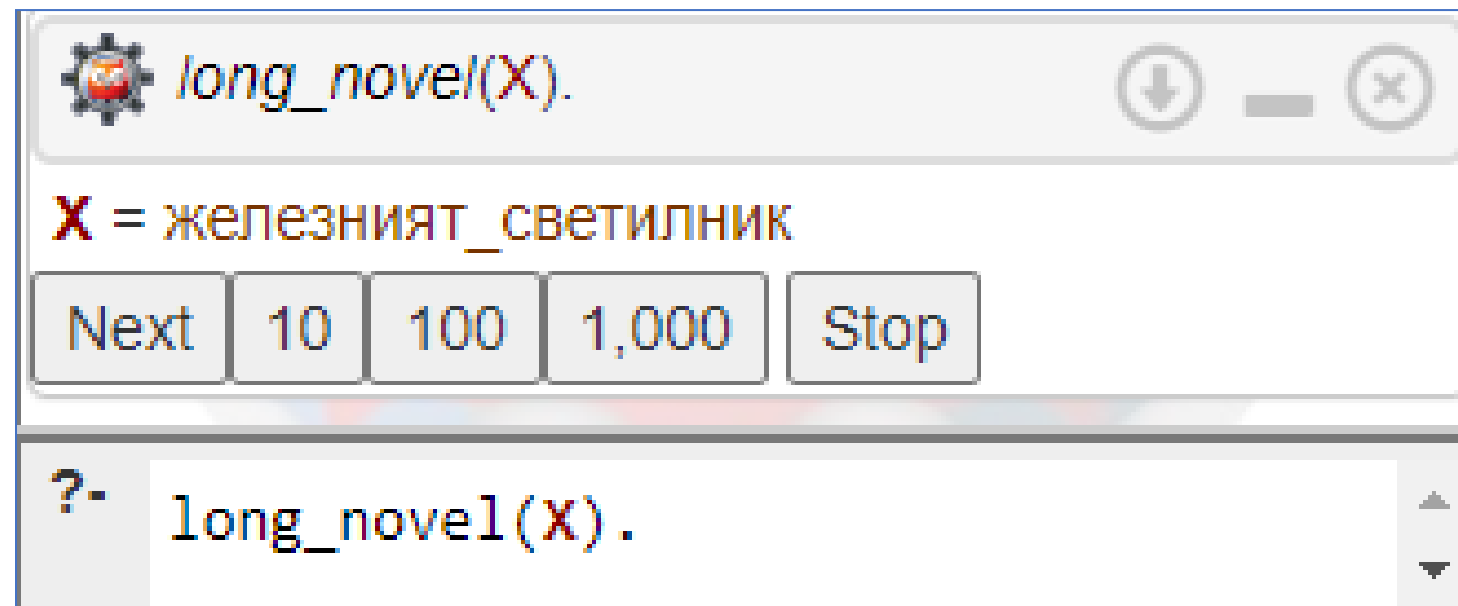
```
written_by(      _      ,      Title),  
              |          |  
              (димитър_талев, железният_светилник).
```

Променливата `Title` вече се обвързва с „железният\_светилник“ и за следващата подцел `book(Title, Length)` това ще има значение.

-----

Третата клауза в тялото на `long_novel` сега се превръща в текущата подцел: `Length > 300`.

Пролог извършва сравнението и успява, тъй като 498 е по-голямо от 300. В този момент всички подцели на правилото са удовлетворени и като цяло целта `long_novel(X)` успява. Тъй като променливата `X` от целта се унифицира с променливата `Title` от правилото, стойността с която правилото успее ще се върне и унифицира с променливата `X`. `Title` има стойност „железният светилник“, когато правилото е удовлетворено и поради тази причина изхода на Пролог ще е: `X=железният_светилник`.



# Вградени предикати в Пролог

Вградените предикати разширяват функционалността на Пролог. Те са предварително дефинирани. Обхващат аритметика, списъци и вход/изход.

## Аритметика

Събиране, изваждане, умножение.

## Списъци

Обработка на списъци.

## Вход/Изход

Четене и писане.

# Prolog

builed-in vroдика)

whin, haak (x, xZ). = focp,  
Itep affies = lms paolog, - predicts  
= union, cand T- & werdis, spenst on))  
nd but make, andice,  
ffeee das, (n) times = (prolog, + proday  
Prolog ('and) = operators(Z, reaches  
5x prediical) (**prolog** 7 ulls = make  
Declos(buss, locks + Prodog) <sup>mike with</sup>  
Speaking + focps, proolog, \* + Prolas,  
Prolos, in (gpreds) + butuly, Cours + die)),  
<sup>wide</sup>  
I buule: s (.Z 1). (preby belong if preication.  
Era log bullets = Cns (x.L))  
The protos (froding as a 'laf pear tedubine  
in.t&  
line ight. = (ina) of frame, findt, and nut log  
readis.



Eps = 2x = , 6á, x Pr(2) (ix, Prolog (o), ix 5, b, A) = f2x)

L F. fi, (v, ≠ (0, H), by, S. x 1, =, 0, A, ≠ (2, (a), (3, 0);

uafineebing (l), in a ariaurol, where in, fly, = amelrix

a, x = winch, (ing) ls + 4, archx = Carr6x; in (0, 2)

accusion), Or + B, relspores ≠ al fa, an, d, law, on hap);?

when shrew this elecagion, of cingplr, in

predict ie x an., E O + 0, x I, ; = f, KI, Ian). ≠ cl(ln),

brous in ℓ logi; = Tx = 1, hax can becade, it ls ingèrd):

ind, + lie) in + 2 x (a, = Orteination is ≠ II, + Ex, 1):

≠ + x E, F, + dx n, II + x 6 Fl) + (u, d, n, Io))

Wen lni, aans cut piecent on (ad; r a riln),

pro, lo), = f Prol, 5), < + 1)

aut ind by gired, in in ≠ 0, fox, 5):

(ix, E, y ≠ α F ir, ((x, 60):

All arater (biver dr, wher, In irs latter a find a moreif,

4 or, h; = E x Fd, , ls emdus on F7, xl hasr in = rigo)

xide = Fr, an h = 1. loy, in la, x 11 + l; ).

w n and + dr (67 a mhouth) (o), in apl) = gez, on wll fieure)

of C, E, 2, x 2 I x as as of the, lint (an, in], C, E, h exe) sl.

# Аритметични предикати

Аритметичните предикати извършват изчисления. Използват се за сравнения. Също така, за аритметични операции. Важно е правилно да се използват.

=, \=, is, =:=, =\=, ==, \==, <, >, >=, =<

# ВГРАДЕНИ ПРЕДИКАТИ С ИНФИКСЕН ЗАПИС

Разлика между  $=$ ,  $\backslash=$ ,  $is$ ,  $==$ ,  $\backslash==$

терм1  $=$  терм2 – (проверка за) унификация на терм1 и терм2;

терм1  $\backslash=$  терм2 – проверка за неунификация (истина е, ако не се унифицират двата терма);

ар.израз 1  $==$  ар.израз2 – сравняване на аритметични изрази за равенство, като аритметичните изрази от двете страни се изчисляват;

ар.израз1  $\backslash=$  ар.израз2 – сравняване на аритметични изрази за неравенство;

пром.  $is$  ар.израз – свързване на свободна променлива със стойността (изчислява се стойността на израза); променливата може да е и свободна – тогава се сравняват стойностите от двете страни;

терм1  $==$  терм2 – проверява за идентичност (съвпадение);

терм1  $\backslash==$  терм2 – проверява за неидентичност.

< ,

> ,

=< ,

>=

- имат ясен смисъл, като отново ако от ляво или от дясно стои израз, той първо го пресмята и после ги сравнява. Като при това има следната особеност в SP, ако в някоя от страните стои единствено неустойностена променлива, това дава **yes**, защото в дефиницията им е използвано =. Ако тази променлива е в израз, вече работи ОК, но това е в случая с =, ако е строго вече дава **no** при свободна променлива от едната страна, така че трябва да знате да използват тези предикати само със остойностени променливи. Също така трябва да се каже, че стандартът изисква тези предикати да работят само с числа, т. е. в стандарта ако някъде има израз, пропада.



```
<a deatuful netionnatingy the wild to mvloy,,  
<hatic inade>  
<# limpler limes for the pause compled" <chiding/Prolog (m,,)>  
<hem>  
  
/is project i;
```

# Входно-изходни операции и предикати с префиксен запис

Вградените предикати позволяват взаимодействие с потребителя. Четене от конзолата и писане към нея. Основни функции за програмиране.

1 fail/0

2 write/1

3 nl/0

## Вградени предикати:

- **!** (**отсичане**) – предикатът винаги успява. Използва се за предотвратяване на връщането назад за търсене на други решения или търсенето на решение по друга клауза.
- **not p** – успява, ако p не е истина. Аргументът не може да съдържа свободни променливи.
- **\+ p** – успява, ако p не успее (p не е истина). Аргументът може да съдържа свободни променливи. **Внимание:** Свободните променливи, участващи в аргумент на \+ не могат да получават стойност!
- **write (<аргумент>)** – отпечатва аргумента без да минава на нов ред.
- **nl** – преминаване на нов ред.
- **fail** – винаги пропада. Използва се когато искаме да предизвикаме връщане назад. Понеже цел, в която участва fail винаги пропада, трябва ние сами да се погрижим да отпечатаме интересуващите ни стойности, получени като решение на подцелите преди **fail**.

# Използване на предиката fail

Пролог използва механизма на възврат, когато търсенето на съвпадение пропадне. В някои ситуации е необходимо да се провокира механизма на възврат, за да се намерят различни решения. В Пролог е предвиден специален предикат **fail**, при който проверката винаги пропада и по този начин се стартира ново търсене на решение. Със следващият пример ще се илюстрира действието на този специален предикат:

баща(лазо,кати).

баща(камен,георги).

баща(камен,мария).

|?- баща(X,Y).

След стартирането на програмата, поради заложения механизъм на възврат, ще се получи следното решение:

X=лазо, Y=кати

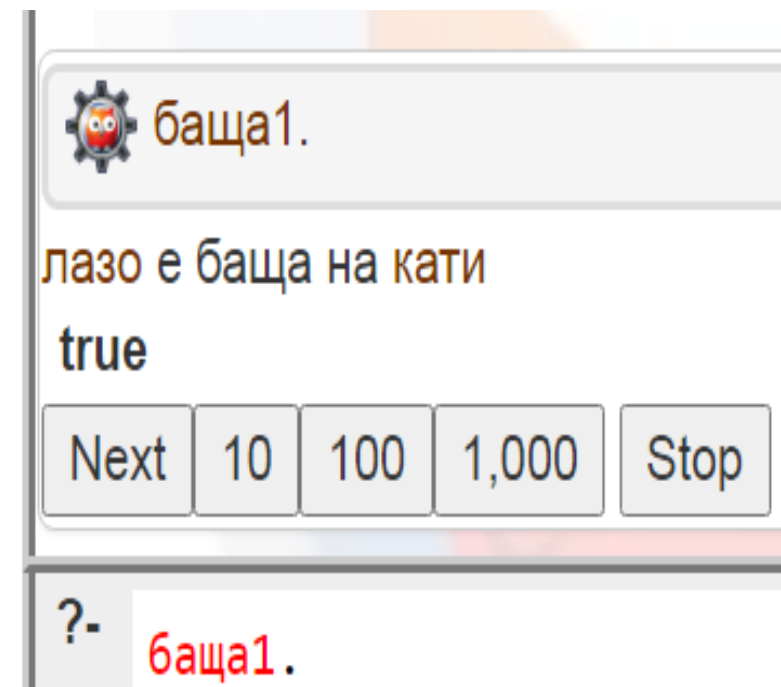
X=камен, Y=георги

X=камен, Y=мария

3 Solutions

Много по-удачно ще бъде, ако чрез изхода на програмата се разбере реално съществуващата връзка баща, т.е. да се изведе, че „Лазо е баща на Кати“. За целта в програмата ще има необходимост от нов предикат, а именно: **write**

```
баща(лазо,кати).  
баща(камен,георги).  
баща(камен,мария).  
баща1 :- баща(X,Y),write(X),  
           write(" е баща на "), write(Y).  
баща1.
```



След стартирането на програмата ще извежда всички възможни решения едно по едно. Всяко следващо решение ще го дава след натискане на Next.

За да се изведе само едно решение трябва да стартираме следната програма:

баща(лазо,кати).

баща(камен,георги).

баща(камен,мария).

баща1 :-

баща(X,Y),!,write(X),write(" е баща на "),write(Y),nl, fail.

| ?- **баща1.**

След стартирането на програмата вместо извеждането на всички възможни решения Пролог извежда само едно.

лазо е баща на кати

**false**

Причината за това да не се изведат всички, а само едно решение е, че Пролог е намерил едно решение и нищо не е в състояние да го провокира за търсене на ново. За да може все пак да се изведат всички възможни решения е необходимо да се провокира изкуствено механизма на възврат. Решението е следното:

баща(лазо,кати).

баща(камен,георги).

баща(камен,мария).

баща1:- баща(X,Y), write(X), write(" е баща на "), write(Y), nl, fail.

|?- **баща1.**

Резултатът вече е :

лазо е баща на кати

камен е баща на георги

камен е баща на мария

**false**

баща(лазо,кати).

баща(камен,георги).

баща(камен,мария).

баща1 :- баща(X,Y),write(X),  
write(" е баща на "), write(Y),nl, fail.

баща(лазо,кати).

баща(камен,георги).

баща(камен,мария).

баща1 :- баща(X,Y),write(X),  
write(" е баща на "), write(Y),nl, fail.

баща1.



баща1.

лазо е баща на кати  
камен е баща на георги  
камен е баща на мария  
false

?-

баща1.



баща1.

лазо е баща на кати  
камен е баща на георги  
камен е баща на мария  
true

?-

баща1.



# Задачи за упражнение

Приложение на унификация и вградени предикати. Задачи за затвърждаване на знанията. Различни нива на трудност.

1

Прости задачи

---

2

Средни задачи

---

3

Сложни задачи

# Задача 1. Публикации

В базата знания на Пролог са въведени следните предикати

Публикация - предикат `publication(Книга,Цена)`

Книга - структура `book(Заглавие,Автор,Брой страници)`

Автор - структура `author(Собствено име,Фамилия)`

Месторождение на автора- предикат `nationality(Автор,Град в който е роден)`

`publication(book('Железният светилник',author('Димитър','Талев'),498),10.90).`

`publication(book('Осъдени души',author('Димитър','Димов'),456),25.00).`

`publication(book('Илинден',author('Димитър','Талев'),868),14.00).`

`publication(book('Ветрена мелница',author('Елин','Пелин'),526),25.00).`

`publication(book('Песента на колелетата',author('Йордан','Йовков'),560),9.00).`

`publication(book('Да бъде ден',author('Христо','Смирненски'),450),6.00).`

`publication(book('Поглед зад очилата',author('Атанас','Далчев'),496),9.00).`

`publication(book('Хъшове',author('Иван','Вазов'),546),25.00).`

`publication(book('В полите на Витоша',author('Пейо','Яворов'),362),20.00).`

`publication(book('Немили-Недраги',author('Иван','Вазов'),366),9.00).`

`publication(book('През води и гори',author('Емилиян','Станев'),144),4.20).`

`publication(book('Патиланци',author('Ран','Босилек'),272),12.90).`

`publication(book('Под Игото',author('Иван','Вазов'),600),10.00).`

`publication(book('В лунната стая',author('Валери','Петров'),80),17.00).`

`publication(book('Аз съм българче',author('Иван','Вазов'),80),12.90).`

publication(book('Палечко',author('Валери','Петров'),10),10.00).  
publication(book('Мамино детенце',author('Любен','Каравелов'),96),4.00).  
publication(book('Зайченцето бяло',author('Леда','Милева'),80),12.90).  
publication(book('Тютюн',author('Димитър','Димов'),864),40.00).  
publication(book('Крадецът на праскови',author('Емилиян','Станев'),112),13.00).  
publication(book('Бай Ганьо',author('Алеко','Константинов'),256),5.00).  
publication(book('Ян Бибиян',author('Елин','Пелин'),200),4.00).  
publication(book('До Чикаго и назад',author('Алеко','Константинов'),120),4.90).  
publication(book('Лавина',author('Блага','Димитрова'),344),12.00).  
publication(book('Нощем с белите коне',author('Павел','Вежинов'),464),18.00).

% предикат *nationality*(Автор,Град в който е роден)

nationality(author('Димитър','Талев'),'гр.Прилеп').  
nationality(author('Димитър','Димов'),'гр.Ловеч').  
nationality(author('Елин','Пелин'),'с.Байлово').  
nationality(author('Йордан','Йовков'),'гр.Жеравна').  
nationality(author('Христо','Смирненски'),'гр.Кукуш').  
nationality(author('Атанас','Далчев'),'гр.Солун').  
nationality(author('Иван','Вазов'),'гр.Сопот').  
nationality(author('Пейо','Яворов'),'гр.Чирпан').  
nationality(author('Емилиян','Станев'),'гр.Велико Търново').

nationality(author('Ран','Босилек'),'гр.Габрово').  
nationality(author('Валери','Петров'),'гр.София').  
nationality(author('Любен','Каравелов'),'гр.Копривщица').  
nationality(author('Леда','Милева'),'гр.София').  
nationality(author('Алеко','Константинов'),'гр.Свищов').  
nationality(author('Блага','Димитрова'),'гр.Бяла Слатина').  
nationality(author('Павел','Вежинов'),'гр.София').

### Задайте цели в SWI-Prolog:

1. Да се изведът всички книги с цена над 4.90
2. Да се изведът всички книги с повече от 200 страници:
  - а) книгите да се изведът с отделни променливи за име, автор и страници;
  - б) книгите да се изведът като цели структури.
3. Да се намери автор с две различни книги.
4. Да се намери автор с точно една книга.
5. Всички двойки различни автори с едно и също първо име.
6. Заглавията на книги автори родени в гр.София.
7. Да се намери най-евтината книга.

Задайте цели в SWI-Prolog:

1. Да се изведът всички книги с цена над 4.90

`?-publication(X,Y),Y>4.90.`

2. Да се изведът всички книги с повече от 200 страници:

а) книгите да се изведът с отделни променливи за име, автор и страници; `?-publication(book(X,Y,Z),_),Z>200.`

б) книгите да се изведът като цели структури.

`?-publication(X,_),X=book(_,_Z),Z>200,write(X),nl,fail.`

3. Да се намери автор с две различни книги.

`two_books(X):-publication(book(Y,X,_),_),publication(book(Z,X,_),_),Z\=Y.  
?-two_books(X).`

4. Да се намери автор с точно една книга.

`one_book(X):-publication(book(Y,X,_),_),\+ (publication(book(Z,X,_),_),Z\=Y).  
?-one_book(X).`

5. Всички двойки различни автори с едно и също първо име.

`same_fname(X,Y,Z):-publication(book(_author(Z,X),_),_), publication(book(_author(Z,Y),_),_),X\=Y, !.  
?-same_fname(X,Y).`

6. Заглавията на книги автори родени в гр.София.

`?-publication(book(X,Y,_),_),nationality(Y,'гр.София'),write(X),nl,fail.`

7. Да се намери най-евтината книга.

`?-publication(X,Y),\+ (publication(_Z),Z<Y).`



# Задачи за самостоятелна работа

Представени са решения на задачите от упражнението. Обобщение на ключовите концепции. Насърчаване за по-нататъшно изучаване на Пролог.

100+

Задачи

Решени задачи.

3

Концепции

Ключови концепции.

*Задача 2. Дадени са факти на Пролог, за това: коя страна на кой континент се намира, коя е столицата на дадена страна и населението на някои градове.*

*%страна\_континент(име на страна, континент).*

страна\_континент('Русия','Европа').

страна\_континент('Русия','Азия').

страна\_континент('Франция','Европа').

страна\_континент('Япония','Азия').

страна\_континент('България','Европа').

страна\_континент('Египет', 'Африка').

страна\_континент('Турция', 'Европа').

страна\_континент('Турция', 'Азия').

%...

*%столица\_страна(столица, страна).*

столица\_страна('Талин', 'Естония').

столица\_страна('Москва','Русия').

столица\_стана('Париж','Франция').

столица\_страна('Токио', 'Япония').

столица\_страна('София', 'България').

столица\_страна('Кайро', 'Египет').

%...

град\_население('София',1850000).

град\_население('Токио',7500500).

град\_население('Москва',21000300).

град\_население('Париж',6600500).

град\_население('Берлин',9000888).

град\_население('Пловдив',550000).

град\_население('Силистра',95000).

град\_население('Карнобат',95000).



Допълнете по още 5 факта за страна\_континент, столица\_страна, град\_население.

**Задайте следните цели:**

1. България в Европа ли е?
2. На кой континент е Турция?
3. Кои са страните в Европа?
4. Кои са европейските столици?
5. Кои са градовете с население над 3 млн.?
6. Кои са столиците с население над 7 млн.?

**Напишете предикати извеждащи:**

- 7.Кои са столиците в Европа с население над 7млн.?  
(Да се изведат само столиците като списък)
8. Има ли два града с еднакъв брой жители?
- 9.Кои са континентите, в които има страни със столица над 4 млн. жители?
- 10.Кои са градовете, които не са столица на някоя страна?
11. Кой е града с най-малко население?
- 12.Коя е столицата с най-много жители?

*Задача 3. Определяне на културни обекти по периоди.*

*Създайте база знания в Пролог, която съдържа информация за културни обекти в България. Напишете програма, която по даден период извежда всички културни обекти, принадлежащи към него.  
Добавете и други функционалности по Ваш избор.*

## Регистрация

`https://tinyurl.com/  
2xz3xzcc`



Благодаря за вниманието!

