

## MECHANIZM WNIOSKOWANIA PROLOGU

### PROGRAM:

```
% Dziadek

/*1*/ojciec(karol, jan) .
/*2*/ojciec(karol, adam) .

/*3*/dziecko(tomasz, jan) .
/*4*/dziecko(piotr, adam) .

/*5*/dziadek(karol, marek) .
/*6*/dziadek(X, Y) :-ojciec(X, Z) , dziecko(Y, Z) .
```

### REALIZACJA CELU: dziadek(karol, X)

<p>Cel 1: <code>dziadek(karol, X)</code></p> <p><code>/*5*/dziadek(karol, marek) .</code></p> <p><math>\sigma</math> : X = marek</p> <p>Cel 1 - zrealizowany</p> <p>Odp: X = marek;</p>	<p>Odszukanie w programie definicji predykatu <code>dziadek</code>.</p> <p>Próba uzgodnienia celu z pierwszą klauzulą definicji <code>dziadek</code>.</p> <p>Uzgodnienie nastąpiło. Klauzula <code>/*5*/</code> zostaje oznaczona specjalnym znacznikiem.</p> <p>Szukanie rozwiązania alternatywnego.</p>
<p>Cel 1: <code>dziadek(karol, X)</code></p> <p><code>/*6*/dziadek(X1, Y1) :-</code>     <code>ojciec(X1, Z1) ,</code>     <code>dziecko(Y1, Z1) .</code></p> <p><math>\sigma</math> : X1 = karol, Y1 = X</p> <p>Cel 2: <code>ojciec(karol, Z1) ,</code>     <code>dziecko(X, Z1) .</code></p> <p>Cel 2.1: <code>ojciec(karol, Z1)</code></p> <p><code>/*1*/ojciec(karol, jan) .</code></p> <p><math>\sigma</math> : Z1 = jan</p> <p>Cel 2.1 - zrealizowany</p> <p>Cel 2.2: <code>dziecko(X, jan)</code></p> <p><code>/*3*/dziecko(tomasz, jan) .</code></p> <p><math>\sigma</math> : X = tomasz</p> <p>Cel 2.2 - zrealizowany</p>	<p>Nawrót do celu ostatnio realizowanego. Zmienna X jest znowu zmienną wolną.</p> <p>Próba uzgodnienia celu z pierwszą klauzulą pod ostatnim znacznikiem definicji <code>dziadek</code>. Ponieważ klauzula zawiera zmienne bierzemy pod uwagę jej wariant. W przypadku reguły cel jest uzgadniany z głową tej reguły.</p> <p>Uzgodnienie nastąpiło. Klauzula <code>/*6*/</code> zostaje oznaczona specjalnym znacznikiem.</p> <p>Aby zrealizować Cel 1 należy zrealizować Cel 2.</p> <p>Najpierw realizowany jest podcel „pierwszy z lewej”. Odszukanie w programie definicji predykatu <code>ojciec</code>.</p> <p>Próba uzgodnienia celu z pierwszą klauzulą definicji <code>ojciec</code>.</p> <p>Uzgodnienie nastąpiło. Klauzula <code>/*1*/</code> zostaje oznaczona specjalnym znacznikiem.</p> <p>Realizacja drugiego podcelu. Odszukanie w programie definicji predykatu <code>dziecko</code>.</p> <p>Próba uzgodnienia celu z pierwszą klauzulą definicji <code>dziecko</code>.</p> <p>Uzgodnienie nastąpiło. Klauzula <code>/*3*/</code> zostaje oznaczona specjalnym znacznikiem.</p>

<p>Cel 1 - zrealizowany</p> <p>Odp: <math>X = \text{tomasz};</math></p>	<p>Szukanie rozwiązania alternatywnego</p>
<p>Cel 2.2: <math>\text{dziecko}(X, \text{jan})</math></p> <p><math>/*4*/\text{dziecko}(\text{piotr}, \text{adam}).</math></p> <p><math>\sigma : X = \text{piotr}, \text{jan} \neq \text{adam}</math></p> <p>Cel 2.1: <math>\text{ojciec}(\text{karol}, Z1)</math></p> <p><math>/*2*/\text{ojciec}(\text{karol}, \text{adam}).</math></p> <p><math>\sigma : Z1 = \text{adam}</math></p> <p>Cel 2.1 - zrealizowany</p> <p>Cel 2.2: <math>\text{dziecko}(X, \text{adam})</math></p> <p><math>/*3*/\text{dziecko}(\text{tomasz}, \text{jan}).</math></p> <p><math>\sigma : X = \text{tomasz}, \text{jan} \neq \text{adam}</math></p> <p><math>/*4*/\text{dziecko}(\text{piotr}, \text{adam}).</math></p> <p><math>\sigma : X = \text{piotr}</math></p> <p>Cel 2.2 - zrealizowany</p> <p>Cel 1 - zrealizowany</p> <p>Odp: <math>X = \text{piotr};</math></p>	<p>Nawrót do celu ostatnio realizowanego. Zmienna X jest znowu zmienną wolną.</p> <p>Próba uzgodnienia celu z pierwszą klauzulą pod ostatnim znacznikiem definicji <b>dziecko</b>.</p> <p>Brak uzgodnienia. Ponieważ nie ma już innych klauzul w definicji <b>dziecko</b>, realizacja Celu 2.2 kończy się porażką i wszystkie znaczniki w tej definicji zostają usunięte.</p> <p>Nawrót do celu wcześniejszego.</p> <p>Próba uzgodnienia celu z pierwszą klauzulą pod ostatnim znacznikiem definicji <b>ojciec</b>.</p> <p>Uzgodnienie nastąpiło. Klauzula <math>/*2*/</math> zostaje oznaczona specjalnym znacznikiem.</p> <p>Realizacja drugiego podcelu. Odszukanie w programie definicji predykatu <b>dziecko</b>.</p> <p>Próba uzgodnienia celu z pierwszą klauzulą definicji <b>dziecko</b>.</p> <p>Brak uzgodnienia.</p> <p>Próba uzgodnienia celu z następną klauzulą definicji <b>dziecko</b>.</p> <p>Uzgodnienie nastąpiło. Klauzula <math>/*3*/</math> zostaje oznaczona specjalnym znacznikiem.</p> <p>Szukanie rozwiązania alternatywnego</p>
<p>Cel 2.2: <math>\text{dziecko}(X, \text{adam})</math></p> <p>Cel 2.1: <math>\text{ojciec}(\text{karol}, Z1)</math></p> <p>Cel 1: <math>\text{dziadek}(\text{karol}, X)</math></p> <p>Odp: false</p>	<p>Nawrót do celu ostatnio realizowanego. Ponieważ nie ma już innych klauzul w definicji <b>dziecko</b>, realizacja Celu 2.2 kończy się porażką i wszystkie znaczniki w tej definicji zostają usunięte.</p> <p>Nawrót do celu wcześniejszego. Ponieważ nie ma już innych klauzul w definicji <b>ojciec</b>, realizacja Celu 2.1 kończy się porażką i wszystkie znaczniki w tej definicji zostają usunięte.</p> <p>Nawrót do celu wcześniejszego. Ponieważ nie ma już innych klauzul w definicji <b>dziadek</b>, realizacja Celu 1 kończy się porażką i wszystkie znaczniki w tej definicji zostają usunięte.</p> <p>Brak innych rozwiązań.</p>

## ŚLEDZENIE WYKONANIA PROGRAMU - PREDYKAT **TRACE/0**.

Prolog umożliwia śledzenie wykonania programu za pomocą predykatu `trace/0`.

Komunikaty podczas śledzenia wykonania programu:

CALL - próba realizacji celu  
EXIT - cel został zrealizowany  
FAIL - próba realizacji celu nie powiodła się  
REDO - ponowienie próby realizacji celu (nawrót)

?- trace.

true

```
[trace] 2 ?- dziadek(karol,X).  
    Call: (7) dziadek(karol, _G466) ? creep  
    Exit: (7) dziadek(karol, marek) ? creep
```

X = marek ;

```
    Redo: (7) dziadek(karol, _G466) ? creep  
    Call: (8) ojciec(karol, _L187) ? creep  
    Exit: (8) ojciec(karol, adam) ? creep  
    Call: (8) dziecko(_G466, adam) ? creep  
    Exit: (8) dziecko(tomasz, adam) ? creep  
    Exit: (7) dziadek(karol, tomasz) ? creep
```

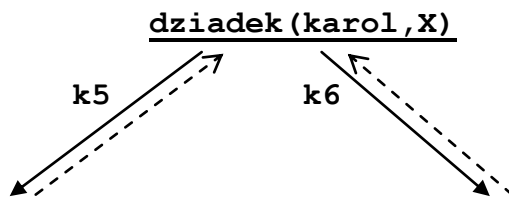
X = tomasz ;

```
    Redo: (8) ojciec(karol, _L187) ? creep  
    Exit: (8) ojciec(karol, jan) ? creep  
    Call: (8) dziecko(_G466, jan) ? creep  
    Exit: (8) dziecko(piotr, jan) ? creep  
    Exit: (7) dziadek(karol, piotr) ? creep
```

X = piotr .

[debug] 3 ?- nodebug.

# DRZEWO POSZUKIWANIA ROZWIĄZANIA



k5: dziadek(karol, marek).

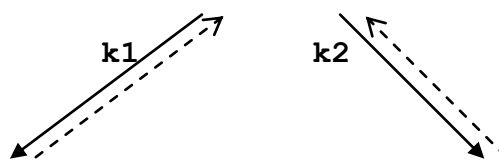
$\sigma: X = \text{marek}$

Odp.  $X = \text{marek};$

wk6: dziadek(X1, Y1) :- ojciec(X1, Z1), dziecko(Y1, Z1)

$\sigma: X1 = \text{karol}, Y1 = X$

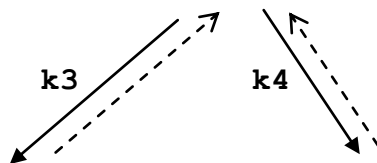
ojciec(karol, Z1), dziecko(X, Z1)



k1: ojciec(karol, jan)

$\sigma: Z1 = \text{jan}$

dziecko(X, jan)



k3: dziecko(tomasz, jan)

$\sigma: X = \text{tomasz}$

Odp.  $X = \text{tomasz};$

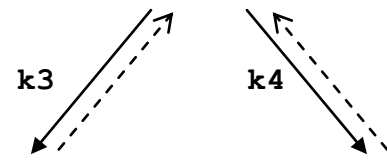
k4: dziecko(piotr, adam)

$\sigma: X = \text{piotr}$   
adam \= jan

k2: ojciec(karol, adam)

$\sigma: Z1 = \text{adam}$

dziecko(X, adam)



k3: dziecko(tomasz, jan) k4: dziecko(piotr, adam)

$\sigma: X = \text{tomasz}$   
jan \= adam

$\sigma: X = \text{piotr}$

Odp.  $X = \text{piotr};$